

Penerapan Algoritma *Dynamic Priority Scheduling* Pada Aplikasi Antrian Pencucian Mobil Berbasis Mobile (Implementation of *Dynamic Priority Scheduling Algorithm* in Mobile Car Queue Washing Applications)

Rizki Setyawati^[1], Adam Bachtiar Maulachela^{[2]*}

^[1] Ilmu Komputer, Universitas Bumigora ^[2] Sekawan Institute

E-mail: rizkisetyawati9@gmail.com, abmaulachela@gmail.com

KEYWORDS:

Dynamic Priority Scheduling Algorithm, Queue, Mobile Application

ABSTRACT

Queue is a condition where the number of service recipients is higher than the number of service providers. This condition can cause a buildup of service recipients, and eventually, bottlenecks will occur. It faced by all service organizations that focus on service to customers. No exception is a car wash business whose business processes focus on providing fast and quality car wash services. But unfortunately, many car wash businesses get complaints from customers, especially related to the queue buildup and unclear queue information received by customers. Therefore this study aims to produce a mobile-based car wash queue application, which includes a dynamic priority scheduling algorithm that functions as a queue manager. To carry out these objectives, a research methodology that is sequential and iterative used, namely, the software development methodology using the Rapid Application Development (RAD) model. This model consists of four phases: planning needs, prototype development, system development, and finally, testing. The Application test is finished with two approaches, namely testing of application code, specifically the application of dynamic priority scheduling algorithm and testing of the overall functional system. From the test results, it knows that the car wash queue application managed to sort customer orders based on the specified priority rules based on the distance and time of the law. While the results of testing the system functionality show that the application successfully manages a variety of errors, both caused by system failures and due to human error.

KATA KUNCI:

Algoritma Penjadwalan Dinamis Berprioritas, Antrian, Aplikasi Mobile

ABSTRAK

Antrian adalah suatu kondisi dimana jumlah penerima layanan lebih besar dibandingkan dengan jumlah pemberi layanan. Kondisi ini dapat mengakibatkan penumpukan penerima layanan dan ujungnya akan terjadi bottleneck. Hal ini bisa dihadapi oleh seluruh organisasi jasa yang fokusnya pada pelayanan kepada pelanggan. Tidak terkecuali usaha pencucian mobil yang proses bisnisnya fokus memberi layanan pencucian mobil yang cepat dan berkualitas. Namun sayangnya usaha pencucian mobil banyak mendapatkan keluhan dari pelanggan khususnya terkait penumpukan antrian dan tidak jelasnya informasi antrian yang diterima pelanggan. Oleh karena itu penelitian ini bertujuan untuk menghasilkan aplikasi antrian pencucian mobil berbasis mobile, yang didalamnya terdapat algoritma dynamic priority scheduling yang berfungsi sebagai pengelola antrian. Untuk melaksanakan tujuan tersebut, maka digunakan metodologi penelitian yang bersifat sequential dan iteratif yaitu metodologi pengembangan perangkat lunak dengan model Rapid Application Development (RAD). Model ini terdiri dari empat fase yaitu perencanaan kebutuhan, pengembangan prototipe, pengembangan sistem, dan yang terakhir pengujian. Pengujian aplikasi dilakukan dengan dua pendekatan yaitu pengujian terhadap kode aplikasi khususnya penerapan algoritma dynamic priority scheduling dan pengujian terhadap keseluruhan fungsional sistem. Dari hasil pengujian diketahui bahwa aplikasi antrian pencucian mobil berhasil mengurutkan pesanan pelanggan berdasarkan pada aturan prioritas yang ditentukan yaitu berdasarkan jarak dan waktu order. Sementara hasil pengujian fungsionalitas sistem menunjukkan aplikasi berhasil mengelola berbagai macam error baik yang diakibatkan oleh kegagalan sistem maupun dikarenakan kesalahan manusia (human error).

I. PENDAHULUAN

Dewasa ini pelaku bisnis di dunia, khususnya di Indonesia mengalami persaingan pasar yang demikian dinamis dan tidak menentu. Persaingan

tersebut sangat terasa, lebih-lebih pada sektor jasa, yang dalam menjalankan bisnisnya setiap pelaku wajib mengedepankan pelayanan yang prima [1].

Oleh karena itu berbagai upaya wajib dilakukan oleh pelaku bisnis di sektor ini.

Salah satunya adalah dengan meningkatkan kualitas layanan agar kepuasan pelanggan meningkat. Tentu saja hal tersebut akan berdampak terhadap peningkatan loyalitas pelanggan. Salah satu bentuk peningkatan kualitas layanan dalam sektor jasa adalah mampu memberikan layanan yang cepat, dengan mempersingkat waktu tunggu layanan.

Waktu tunggu layanan sering disebut sebagai antrian, yaitu keadaan saat penerima layanan menunggu giliran untuk mendapatkan layanan dari pemberi layanan [2]. Antrian timbul sebagai akibat dari tidak seimbangnya perbandingan jumlah penerima layanan dengan pemberi layanan [3].

Salah satu jenis usaha yang bergerak pada sektor jasa adalah pencucian mobil (*car wash*). Jumlah pelaku usaha pada bidang ini meningkat seiring dengan peningkatan jumlah penjualan kendaraan khususnya mobil di suatu daerah. Oleh karena itu usaha pada bidang ini cukup kompetitif, sehingga setiap pelaku usaha berlomba-lomba untuk memberikan layanan yang paling prima.

Namun, masalah utama yang dihadapi hampir seluruh pusat pencucian mobil adalah keluhan pelanggan tentang panjangnya antrian. Selain itu, tidak adanya informasi kapan pelanggan akan mendapatkan layanan. Hal ini tentu saja berdampak pada kekecewaan pelanggan yang ujungnya pelanggan pergi dan mencari tempat lain.

Oleh karena itu, hal yang perlu diperhatikan oleh pemilik jasa layanan pencucian mobil. Mempersingkat waktu layanan agar waktu tunggu menjadi lebih kecil merupakan salah satu solusinya. Namun yang lebih penting lagi adalah kemampuan mengelola antrian. Hal ini perlu diperhatikan karena sebagian besar pelanggan ingin mendapatkan informasi kapan kendaraan mereka akan dilayani.

Berdasarkan pada kendala tersebut, penelitian ini bertujuan untuk menghasilkan aplikasi yang dapat membantu usaha pencucian mobil dalam mengelola antriannya. Pada aplikasi antrian tersebut akan ditanamkan sebuah algoritma penjadwalan berprioritas yang disebut sebagai *dynamic priority scheduling algorithm*. Algoritma ini yang nantinya memiliki peranan untuk mengelola antrian pelanggan berdasarkan pada aturan prioritas yang ditetapkan.

Algoritma ini, adalah pendekatan dinamis dari algoritma penjadwalan berprioritas. Pada

pendekatan dinamis, algoritma ini menitikberatkan proses penentuan antrian berdasarkan pada aturan prioritas yang ditentukan [4]. Berbeda dengan algoritma penjadwalan lainnya yang urutan penjadwalannya fokus pada waktu datang (FCFS), dan pekerjaan terpendek (*Short Job First*) [5]. Pada algoritma ini, setiap proses yang masuk kedalam antrian diidentifikasi berdasarkan nomor prioritas [6]. Kemudian, algoritma ini akan mengatur urutan antrian, mengacu pada nomor prioritas setiap proses, sehingga proses dengan nomor prioritas yang paling tinggi akan mendapatkan urutan antrian pertama.

Namun apabila terdapat proses dengan tingkat prioritas yang sama, maka algoritma akan melihat mana proses yang masuk lebih dahulu masuk kedalam antrian. Artinya algoritma ini, akan mengadopsi algoritma *First Come First Served* (FCFS) sebagai solusinya (Juniarahmatusina, 2014 dalam [6]).

Pemilihan algoritma *dynamic priority scheduling* untuk menyelesaikan masalah antrian antara lain dilakukan oleh beberapa peneliti sebelumnya. Aryawarman, dkk, menerapkan algoritma *dynamic priority scheduling* pada penjadwalan iklan radio Nganjuk FM, dimana aturan prioritas iklan didasarkan pada biaya pemasangan iklan (Aryawarman, 2007 dalam [4]).

Melanjutkan penelitian Aryawarman, Putu, dkk, juga menerapkan algoritma *dynamic priority scheduling* dengan menambahkan aturan prioritas. Pada penelitiannya Putu menggunakan dua parameter yaitu jumlah jam tayang dan lama pemasangan iklan, dengan empat urutan prioritas [4]. Fitri, dkk, pada penelitiannya menggunakan algoritma *priority scheduling* untuk menjadwalkan khutbah Jum'at para mubaliq di IKMI Pekanbaru, dengan tingkat akurasi 93,7 % [6].

Selanjutnya implementasi algoritma penjadwalan CPU juga digunakan oleh Santika, dkk dalam penelitiannya. Dalam penelitiannya itu Santika, dkk, menggunakan algoritma *Shortest Job First* (SJF) dan *Round Robin* (RR) untuk menjadwalkan pengiriman barang [7]. Pada penelitiannya tersebut Santika, dkk berkesimpulan bahwa algoritma SJF lebih baik dalam menjadwalkan pengiriman barang dibandingkan dengan algoritma RR.

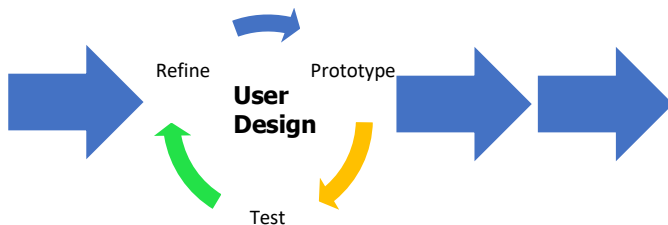
II. METODOLOGI

Penelitian ini dilaksanakan secara sistematis dan bersifat *sequentials* atau berurutan. Selain itu luaran

dari penelitian ini berupa penerapan algoritma *dyanamic priority scheduling* pada aplikasi antrian berbasis *mobile*. Oleh karena itu metodologi yang digunakan pada penelitian ini adalah metodologi untuk pengembangan perangkat lunak.

Model pengembangan perangkat lunak yang digunakan adalah model *Rapid Application Development* (RAD). Penggunaan model pengembangan ini didasarkan pada beberapa aspek seperti waktu pengembangan yang relatif lebih singkat [8], tingginya keterlibatan pengguna (*stakeholders*) dalam pengembangan perangkat lunak [9], dan ruang lingkup yang lebih terdefinisi dan dapat diukur [10].

RAD, memiliki 4 fase utama, agar pengembangan perangkat lunak fokus terhadap kecepatan [11]. Berikut ini merupakan tahapan dari model RAD yang digunakan pada penelitian ini :



Gbr. 1 Fase-fase dalam Model *Rapid Application Development* (RAD) [11].

Seperti yang dijelaskan diatas bahwa model RAD memiliki 4 fase utama, yaitu (1) Perencanaan Kebutuhan (*Requirement Planning*), (2) Pengembangan Prototipe (*User Design*), (3) Pengembangan Sistem (*Construction*), dan (4) Finalisasi fase RAD (*Cutover*).

A. Fase 1 Requirement Planning

Fase ini biasanya disebut sebagai *scope meeting* atau *Joint Application Development* (JAD). Pada fase ini dilakukan diskusi atau *brainstorming* untuk menentukan kebutuhan dari aplikasi atau sistem yang akan dikembangkan. Fase ini akan menghasilkan dokumen kebutuhan sistem yang mencakup ruang lingkup sistem, serta telah disepakati oleh seluruh pemangku kepentingan (*users*).

B. Fase 2 User Design

Fase ini dikenal sebagai tahap pengembangan perangkat lunak. Pada fase ini seluruh kebutuhan diinterpretasikan dalam bentuk *software prototye*.

Kemudian hasilnya diujikan (*test*) kepada pemangku kepentingan untuk di definisikan ulang (*refine*) logika dan/atau alur bisnis yang tidak sesuai.

Hal ini dilakukan sampai seluruh ruang lingkup (*fitur*) perangkat lunak terpenuhi dan disetujui oleh pemangku kepentingan. Proses penerapan algoritma *dynamic priority scheduling* pada aplikasi antrian diterjemahkan pada fase ini.

C. Fase 3 Construction

Fase ini dikenal sebagai *rapid construction*, atau kegiatan mengkonversi hasil protipe pada fase *user design* ke dalam *working model*. Terdapat beberapa kegiatan pada fase ini yaitu : (1) Persiapan *rapid construction*, (2) Pengembangan Program dan Aplikasi, (3) Pengkodean dan Pengujian Unit, (4) Integrasi Sistem.

D. Fase 4 Cutover

Fase terakhir dari model RAD yang digunakan pada penelitian ini adalah *cutover*. Pada fase ini hasil konstruksi aplikasi di uji stabilitasnya. Metode pengujian yang dilakukan adalah dengan *black box* dan *white box* tes. *Black box* tes digunakan untuk menguji fungsionalitas dari aplikasi [12].

Sementara itu pengujian *white box* ditujukan untuk menguji struktur atau alur logika pengkodean [13], khususnya alur logika dari algoritma *dynamic priority scheduling* yang diterapkan pada aplikasi antiran pencucian mobil. Pada fase keempat ini ada satu kegiatan yang tidak dilakukan pada penelitian ini. Kegiatan tersebut adalah perawatan sistem. Perawatan atau *maintenance* tidak dilakukan pada penelitian ini karena pada penelitian ini aplikasi yang dibuat belum digunakan oleh pengguna akhir (*end-users*).

III. HASIL DAN PEMBAHASAN

Untuk dapat menerapkan algoritma *dyanmic priority scheduling* terdapat langkah-langkah prosedur. Dimulai dari penentuan prioritas antrian, penentuan parameter awal antrian, penjadwalan antrian, dan hasil antrian. Berikut ini merupakan pseudocode dari algoritma *dynamic priority scheduling* (DPS) yang diterapkan pada antrian pencucian mobil :

Algoritma Dynamic Priority Scheduling

1. Mulai
2. Ambil data pesanan pelanggan yang berupa, `id_pelanggan`, `plat_no`, `waktu_order`, dan `lokasi_pelanggan`
3. Hitung `jarak_mobil` dengan membandingkan `lokasi_pelanggan` dengan `lokasi_carwash`
4. Bandingkan `jarak_mobil` dari setiap data pesanan yang masuk, kemudian urutkan berdasarkan `jarak_mobil` yang terkecil (KM)
5. Jika terdapat dua atau lebih data_pesanan yang `jarak_mobil` nya sama maka lakukan :
 - a. Ambil `waktu_order` tiap pesanan
 - b. Hitung `menit_pesan` dengan mengurangi `waktu_order` dengan waktu saat ini
 - c. Urutkan pesanan berdasarkan `menit_pesan` yang terbesar
6. Gabungkan pesanan berdasarkan `jarak_mobil` terkecil dan `menit_pesan` terbesar
7. Atur antrian pelanggan
8. Selesai

Terlihat bahwa penerapan algoritma *dynamic priority* memerlukan data utama berupa data pesanan. Pada data pesanan tersebut atribut yang digunakan adalah `jarak_mobil` dan `menit_pesan`. Atribut pertama `jarak_mobil` diambil merupakan hasil hitung dari data koordinat lokasi pencucian mobil yang dibandingkan dengan koordinat lokasi pelanggan melakukan pesanan. Sementara atribut kedua yaitu `menit_pesan`, dihasilkan dari perhitungan waktu saat ini dikurangi dengan `waktu_order` pelanggan yang dikonversi menjadi menit. Setelah kita memahami konsep penerapan algoritma *dynamic priority scheduling* pada kasus antrian pencucian mobil, maka selanjutnya akan ditentukan prioritas antrian.

A. Penentuan Prioritas Antrian

Langkah pertama dalam menerapkan algoritma *dynamic priority scheduling* adalah menentukan prioritas antrian. Terdapat dua tingkat prioritas yang digunakan untuk menentukan urutan antrian pencucian mobil. Kedua tingkat prioritas tersebut

adalah (1) Jarak, dan (2) Waktu Pemesanan, dan diuraikan secara detail pada Tabel I dibawah ini :

TABEL I
PRIORITAS ANTRIAN PENCUCIAN MOBIL

Urutan Prioritas	Penjelasan
Prioritas 1	User atau pelanggan dengan jarak paling dekat dengan lokasi tempat pencucian mobil
Prioritas 2	Jika terdapat 2 atau lebih <i>user</i> atau pelanggan yang lokasinya sama, maka urutan pelanggan didasarkan pada <i>user</i> atau pelanggan mana yang terlebih dulu order (mengacu pada <i>datetime</i>) adalah <i>user</i> yang mendapat prioritas lebih tinggi.

Setelah ditentukan urutan prioritas antrian pencucian mobil, berikutnya ditentukan parameter untuk menentukan ambang batas dalam penentuan antrian pencucian mobil.

B. Penentuan Parameter Awal

Terdapat dua parameter untuk menentukan batasan prioritas dalam penerapan algoritma *dynamic priority scheduling*, yaitu :

1) Jarak Kendaraan > 500 meter

Apabila terdapat dua kendaraan (mobil A dan mobil B) sama-sama berjarak lebih dari 500 meter dari lokasi pencucian mobil. Mobil A memiliki jarak 2 km, sementara mobil B 3,5 km. Kemudian keduanya melakukan order, maka mobil dengan jarak terdekat (mobil A) mendapatkan urutan lebih dahulu dibandingkan dengan mobil B.

2) Jarak Kendaraan (≤ 500 meter) dan Waktu Order

Apabila terdapat dua kendaraan (mobil A dan mobil B) sama-sama berjarak kurang dari 500 meter dari lokasi pencucian mobil. Mobil B lebih dahulu melakukan order dibandingkan dengan mobil A, maka, mobil dengan waktu order lebih dahulu (mobil B) mendapatkan prioritas yang lebih tinggi dibandingkan dengan mobil A. Berdasarkan uraian diatas terlihat bahwa parameter waktu order hanya akan aktif jika terdapat kendaraan yang berada pada jarak kurang dari sama dengan 500 meter dari lokasi pencucian mobil.

C. Penjadwalan Antrian

Jika diasumsikan dalam aplikasi pencucian mobil menerima 4 order pencucian mobil dengan detail pesanan ditampilkan pada Tabel II dibawah ini

TABEL II
DETAIL ORDER PENCUCIAN MOBIL

Pelanggan	Jarak	Waktu Order
Alisya	3 Km	15:24:02
Setyawati	2 Km	15:24:01
Rizki	≤500 meter	15:24:03
Dedy	≤500 meter	15:25:00

Terlihat pada Tabel II di atas terdapat dua kelompok pelanggan, yaitu pelanggan dengan jarak > 500 meter (*Alisya dan Setyawati*) dan pelanggan dengan jarak ≤ 500 meter (*Rizki dan Dedy*). Dari kedua kelompok diatas dapat dipastikan kelompok pertama (*Alisya dan Setyawati*) akan mendapat urutan prioritas dibawah kelompok kedua (*Rizki dan Dedy*).

Namun disetiap kelompok belum bisa ditentukan mana yang memiliki prioritas lebih tinggi. Untuk itu digunakan parameter yang sudah ditentukan diatas. Pada kelompok pertama (> 500 meter), digunakan parameter jarak. Terlihat bahwa jarak *Setyawati* (2 km) lebih dekat dibandingkan dengan *Alisya* (3 km). Oleh karena itu *Setyawati* berada pada urutan 3 dan *Alisya* berada pada urutan 4.

Sementara itu pada kelompok kedua (≤ 500 meter), digunakan parameter jarak dan waktu order. Hal ini karena kedua pelanggan berada pada jarak kurang dari 500 meter. Pada kondisi ini aplikasi antrian secara otomatis akan mengaktifkan parameter waktu order. Oleh karena itu *Rizki* berada pada urutan 1 sementara *Dedy* pada urutan prioritas ke-2. Hal ini dikarenakan waktu order *Dedy* (15:25:00) berada dibawah waktu order *Rizki* (15:24:03).

D. Hasil Penjadwalan

Berikut ini merupakan hasil penjadwalan antrian pencucian mobil berdasarkan pemesanan yang dilakukan melalui aplikasi.

TABEL III
HASIL PENJADWALAN ANTRIAN

Pelanggan	Jarak	Waktu Order	Urutan Prioritas
Rizki	≤ 500 meter	15:24:03	1
Dedy	≤500 meter	15:25:00	2
Setyawati	2 Km	15:24:01	3
Alisya	3 Km	15:24:02	4

Berdasarkan hasil perhitungan oleh algoritma *dyanamic priority scheduling* diperoleh urutan prioritas pelanggan yaitu Rizki, Dedy, Setyawati, dan terakhir Alisya.

E. Hasil Pengujian

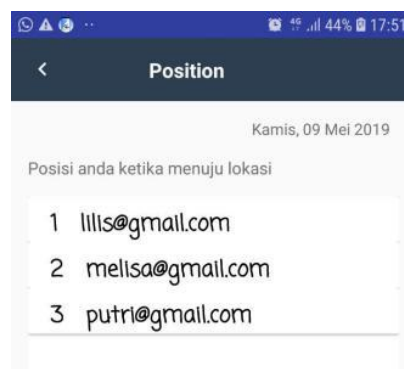
Pengujian yang dilakukan pada penelitian ini terdiri dari dua pendekatan. Pendekatan pertama adalah pengujian yang dilakukan terhadap algoritma *dynamic priority scheduling* yang diterapkan pada aplikasi antrian pencucian mobil, dengan metode *white box*. Sementara itu pengujian yang kedua adalah menguji fungsionalitas sistem dengan menggunakan metode *black box*.

TABEL IV
HASIL UJI COBA PENJADWALAN ANTRIAN KETIKA JARAK SELURUH PELANGGAN YANG ORDER SAMA

Pelanggan	Jarak	Waktu Order	Urutan Prioritas
lilis@gmail.com	2,5 Km	17.46.21	1
melisa@gmail.com	2,5 Km	17.47.45	2
putri@gmail.com	2,5 Km	17.50.09	3

Sumber : Hasil Output Aplikasi Antrian

Pada Tabel IV terlihat, hasil pengujian terhadap penerapan algoritma *dynamic priority scheduling* ketika pelanggan yang memesan berada pada jarak yang sama. Aplikasi menjalankan algoritma *dynamic priority scheduling* dengan mengubah prioritas urutan berdasarkan waktu order yang masuk sebagaimana terlihat pada Gbr. 2 di bawah ini.



Gbr. 2 Hasil Urutan Prioritas Pelanggan Ketika Jarak Seluruh Pelanggan Sama.

Berikutnya ketika ada pelanggan yang melakukan order dengan jarak yang lebih dekat dari ketiga pelanggan sebelumnya, maka urutan prioritas akan berganti. Tabel V dibawah ini akan menampilkan perubahan urutan prioritas ketika kasus diatas terjadi.

TABEL V
HASIL UJI COBA PENJADWALAN ANTRIAN KETIKA JARAK SELURUH PELANGGAN YANG ORDER BERBEDA

Pelanggan	Jarak	Waktu Order	Urutan Prioritas
lilis@gmail.com	2,5 Km	17.46.21	2
melisa@gmail.com	2,5 Km	17.47.45	3

putri@gmail.com	2,5 Km	17.50.09	4
dea@gmail.com	2 Km	17.55.49	1

Sumber : Hasil Output Aplikasi Antrian

Terlihat pada Tabel 5, ada pelanggan yang melakukan pemesanan. Jarak pelanggan baru (dea@gmail.com) memiliki jarak yang lebih dekat dengan tempat pencucian dibandingkan 3 pelanggan sebelumnya. Oleh karena itu aplikasi akan mengubah urutan prioritasnya sesuai dengan jarak pelanggan yang melakukan pemesanan, dan tidak memperdulikan lagi waktu order pelanggan.

Selanjutnya dilakukan pengujian terhadap fungsionalitas dari aplikasi antrian pencucian mobil. Pengujian dilakukan dengan menggunakan metode *black box*. Desain tes menghasilkan skenario tes terhadap 8 *case*, yaitu (1) Registrasi pelanggan, (2) Login, (3) Penambahan layanan, (4) Keranjang layanan, (5) Order (Pemesanan), (6) Master data, (7) Posisi pelanggan, (8) Antrian layanan.

Terdapat 2 jenis skenario yang dibuat dalam pengujian *black box*. Yang pertama menguji dengan memasukkan data yang benar, dan yang kedua menguji dengan memasukkan data yang salah. Kedua jenis skenario ini memiliki tujuan yang berbeda. Jenis yang pertama untuk menguji apakah sistem akan menghasilkan luran yang diharapkan atau tidak. Sementara jenis yang kedua fokus pada apakah sistem dapat menangani *error, bug*, maupun *defect* jika data yang dimasukkan bukan seperti yang diharapkan [14][15].

TABEL VI
HASIL TEST DARI CASE ANTRIAN LAYANAN

ID	Skenario	Test Case	Hasil Yang Diharapkan	Hasil Percobaan	Status
AL 01	User Buka halaman l antrian layanan	Klik tombol antrian	Sistem secara otomatis menampilkan daftar antrian pelanggan	Aplikasi menampilkan daftar antrian pelanggan	diterima
AL 02	Matikan koneksi internet, dan User Buka halaman layanan antrian	Klik tombol antrian	Sistem memberikan pesan error, "List Tidak Dapat Dibuka, Cek Koneksi Anda !!!"	Aplikasi menampilkan error dengan pesan "List Tidak Dapat Dibuka, Cek Koneksi Anda !!!"	diterima

Sumber : Hasil Pengujian Fungsional Aplikasi

Terlihat pada Tabel VI diatas terdapat dua skenario pengujian fungsi menu antrian layanan dari aplikasi. Pada pengujian pertama skenario uji memerintahkan user agar membuka halaman antrian layanan. Hasil yang diharapkan aplikasi menampilkan daftar antrian pelanggan. Setelah dilakukan pengujian didapatkan bahwa sistem menampilkan daftar antrian pelanggan, seperti Gbr. 2 di atas.

Sementara itu pada skenario kedua user diminta melakukan skenario yang sama dengan skenario pertama. Namun pada skenario kedua koneksi internet dari *smartphone* dimatikan. Maka ketika user membuka menu antrian layanan akan muncul pesan *error* "List Tidak Dapat Dibuka, Cek Koneksi Internet Anda !!!".

Berdasarkan dua jenis skenario diatas, hasil pengujian sistem menunjukkan bahwa aplikasi mampu menghasilkan luaran yang diharapkan dan juga mampu menanggapi *error, bug*, dan *defact* yang dihasilkan oleh kesalahan sistem ataupun yang disebabkan oleh kesalahan manusia (*human error*).

IV. PENUTUP

Algoritma *dynamic priority scheduling* berhasil diterapkan pada aplikasi pencucian mobil. Hal ini terbukti dengan dihasilkannya urutan prioritas pelanggan berdasarkan pada aturan prioritas yaitu jarak dan waktu order pada aplikasi. Dengan menanamkan algoritma *dynamic priority scheduling* urutan antrian menjadi dinamis (berubah-ubah seiring dengan masukkan pesanan) disesuaikan dengan aturan prioritas yang ditetapkan. Hasil uji sistem baik dengan pendekatan *white box* dan *black box* menunjukkan implementasi algortima *dynamic priority scheduling* berjalan dengan baik, dan seluruh fungsionalitas aplikasi bekerja dengan baik.

Penelitian ini masih menggunakan parameter jarak dan waktu order. Untuk penelitian ke depan bisa ditambahkan parameter lainnya sebagai dasar penentuan prioritas. Salah satu yang bisa ditambahkan sebagai parameter adalah *membership*, ataupun frekuensi pelanggan menggunakan layanan.

UCAPAN TERIMA KASIH

Para peneliti pada aritkel ini mengucapkan terima kasih kepada para pihak yang telah membantu. Khususnya PT. Lamone Rangga Putra sebagai lokasi studi kasus. Selain itu juga kepada Univesitas

Bumigora dan Lembaga Penelitian Sekawan Institute.

REFERENSI

- [1] D. Prayogo, J. Pondaag, and F. Ferdinand Tumewu, "Analisis Sistem Antrian Dan Optimalisasi Pelayanan Teller Pada PT. Bank Sulutgo," *J. Ris. Ekon. Manajemen, Bisnis dan Akunt.*, vol. 5, no. 2, pp. 928–934, 2017.
- [2] S. Harahap, U. Sinulingga, and S. Ariswoyo, "Analisis Sistem Antrian Pelayanan Nasabah Di Pt. Bank Negara Indonesia (Persero) Tbk Kantor Cabang Utama Usu," *Saintia Mat.*, vol. 2, no. 3, pp. 277–287, 2014.
- [3] S. N. Aulele, "Analisis Sistem Antrian Pada Bank Mandiri Cabang Ambon," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 8, no. 1, pp. 45–49, 2014.
- [4] P. S. Oktaviany, A. B. Maulachela, A. Ashril Rizal, and A. S. Anas, "Sistem Penjadwalan Iklan Radio Menggunakan Algoritma Dynamic Priority Scheduling Berbasis Web (Studi Kasus: Radio Global FM Lombok)," 2018, no. Selisik, pp. 393–399.
- [5] A. P. U. Siahaan, "Comparison Analysis of CPU Scheduling : FCFS, SJF and Round Robin," *Int. J. Eng. Dev. Res.*, vol. 4, no. 3, pp. 124–131, 2016.
- [6] A. dkk Fitri, "Penerapan Constraint Satisfaction Problem pada metode Priority Scheduling untuk Penjadwalan Khutbah Jum'at para Mubaligh di IKMI Pekanbaru," *Sains*, vol. 13, no. 2, pp. 190–194, 2016.
- [7] M. Santika and S. Hansun, "Implementasi Algoritma Shortest Job First dan Round Robin pada Sistem Penjadwalan Pengiriman Barang," *J. Ultim.*, vol. 6, no. 2, pp. 94–99, 2014.
- [8] S. Aswati *et al.*, "Studi Analisis Model Rapid Application Development Dalam," *Stud. Anal. Model Rapid Appl. Dev. Dalam*, vol. 16, no. 3, p. 2, 2017.
- [9] J. R. Sagala, "Model Rapid Application Development (Rad) Dalam Pengembangan Sistem Informasi Penjadwalan Belajar Mengajar," *J. Mantik Penusa*, vol. 2, no. 1, pp. 87–90, 2018.
- [10] H. T. Hidayat, "Pengujian Kualitas Kelayakan Perangkat Lunak Dengan Penerapan perancangan Model Rapid Application Development," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 2, no. 2, pp. 121–129, 2017.
- [11] M. Tajuddin, A. B. Maulachela, Ismarmiaty, N. K. Sriwinarti, A. Juliansyah, and A. Ashril Rizal, "Pengembangan Sistem Informasi," in *Sistem Informasi*, 1st ed., Yogyakarta: Deepublish, 2020, pp. 70–73.
- [12] C. Henard, M. Papadakis, M. Harman, and Y. Le Traon, "Comparing White-box and Black-box Test Prioritization," pp. 523–534, 2016.
- [13] M. E. Khan, "Different approaches to white box testing technique for finding errors," *Int. J. Softw. Eng. its Appl.*, vol. 5, no. 3, pp. 1–14, 2011.
- [14] H. Bhasin, E. Khanna, and S. Sudha, "Black Box Testing based on Requirement Analysis and Design Specifications," *Int. J. Comput. Appl.*, vol. 87, no. 18, pp. 36–40, 2014.
- [15] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, 2012.