# Implementasi Algoritma *Priority Scheduling* untuk Optimalisasi Proses *Loading In* dan *Loading Out* pada Vendor Event di Kalimantan Tengah (ABC Music)

**Samuel Chrysostomus Jaya Gasan[1)], Licantik[2)], Agus Sehatman Saragih[3)]**

[1)2)3)]Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Kampus Tunjung Nyaho Jalan Yos Sudarso, Palangka Raya, Kalimantan Tengah, Indonesia
[1)]gasanem088@gmail.com
[2)]licantik@it.upr.ac.id
[3)]assaragih@it.upr.ac.id

**Abstrak**

Penjadwalan proses *Loading In* dan *Loading Out* pada *vendor Event Organizer* sering mengalami kendala seperti bentrokan jadwal, konflik kru, dan keterbatasan *inventori*. Proses manual membuat pengelolaan *event* menjadi kurang efektif dan memerlukan waktu lebih lama. Penelitian ini bertujuan membangun sistem penjadwalan *event* berbasis *web* menggunakan algoritma *Priority Scheduling* untuk menentukan prioritas event secara otomatis dan terstruktur. Sistem dikembangkan menggunakan metode Waterfall dengan bahasa pemrograman Python, Framework Flask, dan Database MySQL. Penentuan prioritas dilakukan berdasarkan parameter jarak lokasi, durasi event, biaya event, kompleksitas barang, dan jumlah kru yang menghasilkan nilai *Priority Score* untuk pengurutan jadwal *event*. Hasil penelitian menunjukkan bahwa sistem mampu melakukan penjadwalan *event* secara otomatis, menampilkan prioritas *Loading In* dan *Loading Out*, mendeteksi konflik kru dan *inventori*, serta membantu *Monitoring* kegiatan *Event Organizer*. Berdasarkan pengujian *Black Box Testing*, seluruh fitur berjalan dengan baik dan penerapan algoritma *Priority Scheduling* meningkatkan efisiensi proses penjadwalan sebesar 40% dibandingkan metode manual..

**Kata kunci***: Priority Scheduling, Event Organizer, Loading In, Loading Out, Flask*, Penjadwalan *Event*

***Abstract***

*The Scheduling process of Loading In and Loading Out in Event Organizer vendors often faces problems such as schedule conflicts, crew conflicts, and limited inventory availability. Manual Scheduling makes event management less effective and time-consuming. Therefore, this study aims to develop a web-based event Scheduling system using the Priority Scheduling algorithm to determine event priorities automatically and systematically. The system was developed using the Waterfall method with Python programming language, Flask Framework, and MySQL Database. Event priorities are determined based on several parameters, including event location distance, event duration, event cost, item complexity, and the number of crew members, which produce a Priority Score for Scheduling purposes. The results show that the system successfully performs automatic event Scheduling, Displays Loading In and Loading Out priorities, detects crew and inventory conflicts, and supports Event Organizer activity Monitoring. Based on Black Box Testing, all system features function properly according to User requirements. In addition, the implementation of the Priority Scheduling algorithm improved Scheduling efficiency by 40% compared to the manual method.*

***Keywords:*** *Priority Scheduling, Event Organizer, Loading In, Loading Out, Flask, Event Scheduling*

DOI:https://doi.org/10.47111/jointecoms.v6i3 *Received: 01-09-2026*
*Accepted: 10-09-2026*

## 1. PENDAHULUAN

Dalam dunia industri *event*, manajemen waktu dan sumber daya menjadi kunci keberhasilan pelaksanaan acara. Proses *Loading In* dan *Loading Out* adalah tahapan vital di mana semua peralatan harus dikirim dan dikembalikan secara tepat waktu. Di Kalimantan Tengah, ABC Music sebagai *vendor event* sering menghadapi tantangan seperti keterlambatan (Ban Bocor, Pengisian Bahan Bakar, Ban Mobil Terperosok Lubang, Dan Tanah yang tidak seimbang mengakibatkan alat angkut miring ke kiri dan kanan).

Berdasarkan kondisi tersebut, dibutuhkan pendekatan sistematis untuk mengelola penjadwalan dengan lebih optimal. Algoritma *Priority Scheduling* dipilih karena mampu mengatur urutan proses berdasarkan tingkat kepentingan dan urgensi. Algoritma ini telah terbukti efektif dalam konteks operasional lain, seperti sistem antrian dan pelayanan publik. Penerapannya dalam konteks logistik acara diharapkan dapat meningkatkan efisiensi dan mengurangi konflik jadwal, sekaligus memberikan dasar teknis yang terukur.

Salah satu metode yang dapat digunakan untuk mengoptimalkan manajemen proyek adalah Algoritma *Priority Scheduling*. Algoritma ini berfungsi untuk mengatur urutan dan efisiensi proses *Loading In* dan *Loading Out* berdasarkan tingkat urgensi dan jenis peralatan. Dengan menerapkan Algoritma *Priority Scheduling*, diharapkan ABC Music dapat meningkatkan efisiensi dalam proses *Loading In* dan *Loading Out*, sehingga dapat mengurangi waktu yang dibutuhkan dan meningkatkan kepuasan klien.

## 2. TINJAUAN PUSTAKA

### 2.1 ABC *Music*

ABC Music Merupakan Salah Satu Penyedia Jasa *Event Profesional* yang ada di Kota Palangka Raya, Kalimantan Tengah. Mereka menawarkan layanan lengkap untuk berbagai kebutuhan acara, termasuk: *Sound System, Lighting*, Layar LED Videotron, Genset, *Backdrop* Dan *Rigging*. Di Abc Music terdapat Divisi Logistik dan Divisi Operasional yang berperan penting dalam progres *Loading In* dan Loading Out barang.

### 2.2 Konsep Event Organizer

*Event Organizer* adalah entitas atau individu yang bertanggung jawab untuk merencanakan, mengorganisir, dan melaksanakan acara. Tugas utama mereka mencakup pengelolaan semua aspek acara, mulai dari pemilihan lokasi, pengaturan logistik, hingga koordinasi dengan berbagai pihak terkait. Dalam konteks ini, proses *Loading In* dan *Loading Out* menjadi bagian penting dari manajemen logistik yang harus dikelola dengan baik untuk memastikan kelancaran acara.

### 2.3 Proses *Loading In* dan *Loading Out*

Proses *Loading In* merujuk pada kegiatan memindahkan barang, peralatan, dan perlengkapan ke lokasi acara sebelum acara dimulai. Sebaliknya, *Loading Out* adalah proses mengeluarkan barang-barang tersebut setelah acara selesai. Kedua proses ini memerlukan perencanaan yang matang untuk menghindari keterlambatan dan memastikan semua barang tersedia tepat waktu.

### 2.4 Algoritma Penjadwalan

Penjadwalan adalah proses menentukan urutan tugas atau pekerjaan yang harus dilakukan dalam waktu tertentu. Dalam konteks *Loading In* dan *Loading Out*, algoritma penjadwalan yang tepat dapat meningkatkan efisiensi dan mengurangi waktu tunggu.

### 2.5 *Priority Scheduling*

*Priority Scheduling* adalah algoritma penjadwalan yang memberikan prioritas kepada tugas tertentu berdasarkan kriteria yang telah ditentukan. Dalam konteks *Event Organizer*, prioritas dapat ditentukan berdasarkan urgensi, jenis barang, atau waktu kedatangan.

## 2.6 Penjadwalan

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas, serta pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada (Nunung Indra Lesmana, 2016).

## 2.7 *Flowchart*

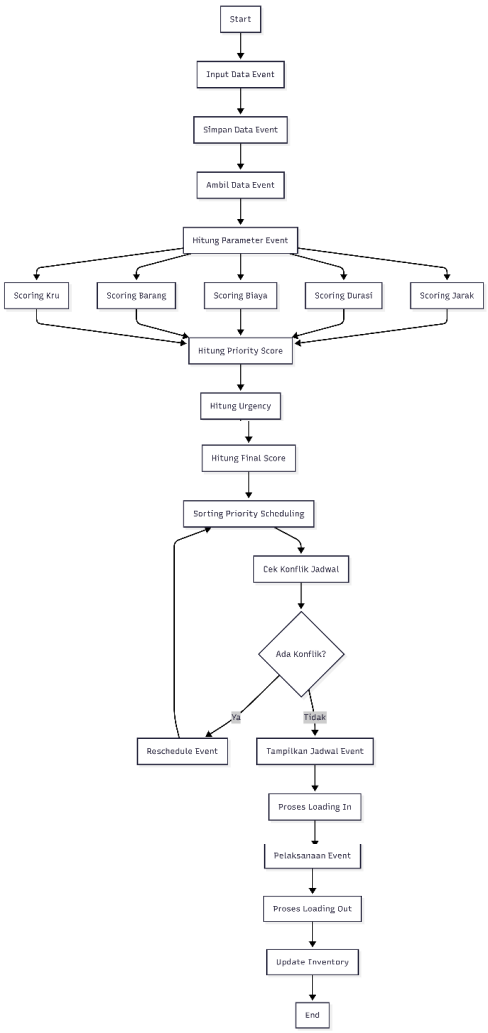
*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses dengan proses lainnya dalam suatu program. Biasanya, seorang analis untuk sistem menggunakan *Flowchart* sebagai bukti dokumentasi menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *Programmer*.

# 3. METODE PENELITIAN

## 3.1 Analisis

Untuk memperoleh data yang diperlukan dalam penelitian ini, penulis menggunakan metode wawancara (*Interview*) dengan bertanya kepada *owner* ABC Music. Analisis yang dilakukan berupa analisis *system* lama atau yang sedang berjalan. Sehingga nantinya dapat diketahui *system* baru seperti apa yang diperlukan. Analisis sistem meliputi bisnis proses dan *Flowchart*.

### 3.1.1 *Flowchart* Sistem




Gambar 1. *Flowchart* system *Loading In* dan *Loading Out*

### 3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Waterfall Model.* Metode *Waterfall* dipilih karena memiliki tahapan pengembangan yang terstruktur dan sistematis sehingga memudahkan proses perancangan dan pembangunan sistem penjadwalan event.

#### 3.2.1. Tahapan Metode *Waterfall*

Tahapan metode *Waterfall* yang digunakan dalam penelitian ini meliputi:

1. Analisis Kebutuhan
2. Perancangan Sistem
3. Implementasi Sistem
4. Pengujian Sistem

### 3.3 Desain *Database*

Tahapan Desain *Database* yang digunakan dalam penelitian ini meliputi:

1. Desain *Entity Relationship Diagram* (ERD)
2. Desain Tabel
3. Relasi Antar Tabel
4. *System And Software Design*
5. Contoh Perhitungan *Priority Scheduling*
6. Struktur *Database*
7. Metrik Pengukuran Peningkatan

## 4. PEMBAHASAN

### 4.1 Implementasi Sistem

Pada penelitian ini dilakukan implementasi sistem penjadwalan *event* berbasis *web* yang digunakan untuk membantu *vendor event* dalam mengatur jadwal kegiatan seperti proses *Loading In*, pelaksanaan *event*, dan *Loading Out*. Sistem ini dikembangkan menggunakan bahasa pemrograman *Python* dengan *Framework Flask*, serta menggunakan MySQL sebagai sistem manajemen basis data.

Sistem ini dirancang untuk membantu *vendor* dalam menentukan prioritas *event* dengan menggunakan algoritma *Priority Scheduling*. Algoritma ini menentukan urutan prioritas *event* berdasarkan beberapa parameter yang mempengaruhi tingkat kepentingan suatu *event.*

Beberapa parameter yang digunakan dalam perhitungan prioritas antara lain:

1. Jarak lokasi *event*,
2. Durasi *event*,
3. Biaya *event*,
4. Jumlah dan tingkat kesulitan peralatan,
5. Jumlah kru yang dibutuhkan.

Sistem akan menghitung nilai prioritas berdasarkan parameter tersebut, kemudian menentukan *event* mana yang memiliki prioritas lebih tinggi

### 4.2 Implementasi Antarmuka Sistem

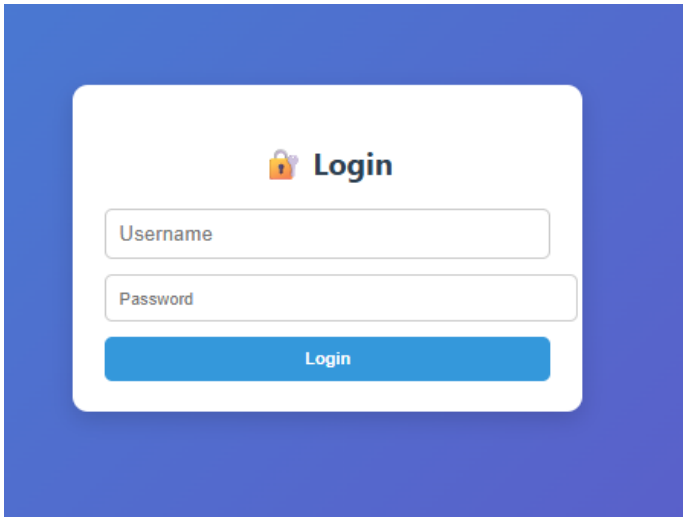
#### 4.2.1 Halaman *Login*

Halaman *login* digunakan sebagai proses autentikasi pengguna sebelum masuk ke dalam sistem. Pengguna harus memasukkan *User*name dan *password* yang sesuai agar dapat mengakses *fitur* sistem.

Pada halaman ini terdapat :

1. form *User*name,
2. form *password*,
3. serta tombol *login.*

Sistem akan melakukan validasi data pengguna sebelum mengarahkan pengguna ke halaman utama sistem.




Gambar 2. Halaman *Login*

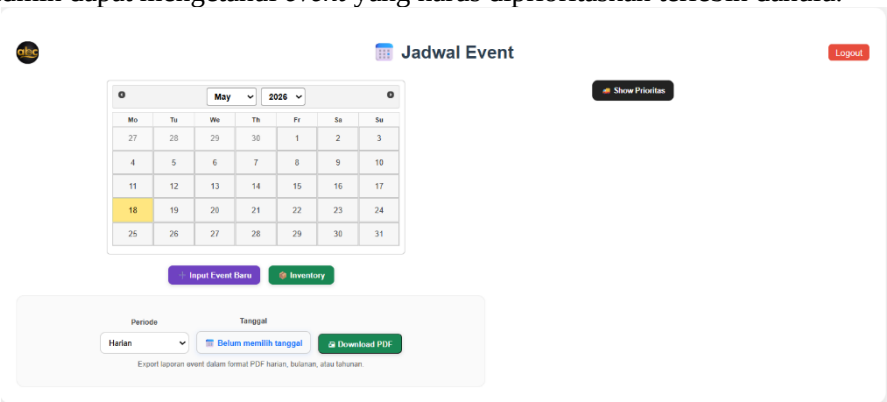
### 4.2.2 Halaman Dashboard Jadwal Event

Halaman *dashboard* merupakan halaman utama sistem yang digunakan untuk menampilkan kalender jadwal *event* serta ringkasan proses *Scheduling Event Organizer*.

Pada halaman ini admin dapat :

1. melihat kalender *event*,
2. melihat daftar *event* berdasarkan tanggal,
3. melihat prioritas *Loading In* dan *Loading Out*,
4. mengakses menu *inventori*,
5. menambahkan *event* baru,
6. serta melakukan *Export* laporan PDF.

Sistem juga menampilkan daftar prioritas *event* berdasarkan nilai *Priority Score* tertinggi sehingga admin dapat mengetahui *event* yang harus diprioritaskan terlebih dahulu.




Gambar 3. Halaman *Dashboard* Jadwal *Event*

### 4.2.3 Halaman *Input* Event

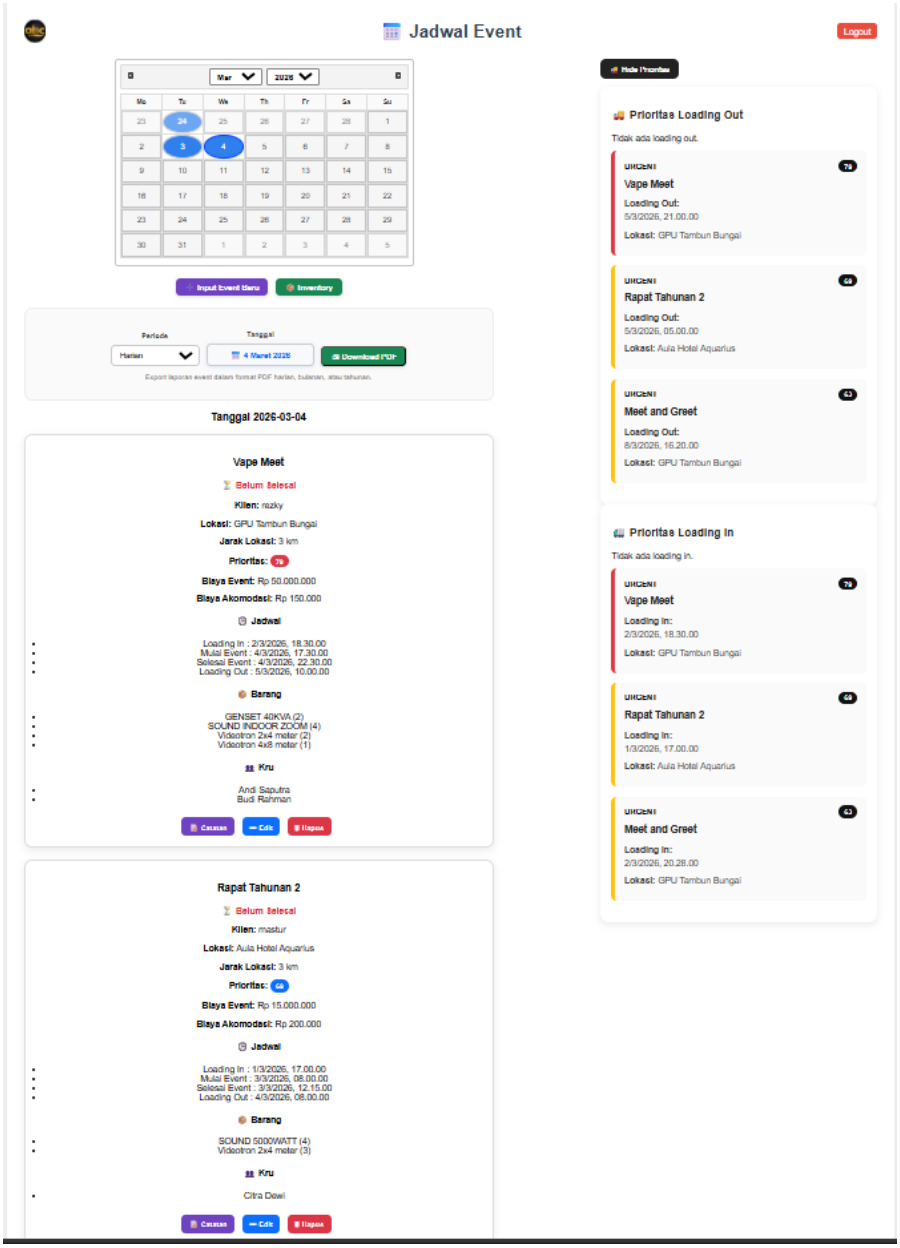
Halaman *Input* event digunakan untuk menambahkan data event baru ke dalam sistem.

Pada halaman ini admin dapat mengisi :

1. nama *event*,
2. nama klien,
3. lokasi *event*,
4. biaya akomodasi,
5. jarak lokasi,
6. tanggal dan waktu *event*,
7. jadwal *Loading In*,
8. jadwal *Loading Out*,
9. durasi *event*,

10. biaya *event*,
11. barang yang digunakan,
12. serta kru yang bertugas.

Setelah *data event* disimpan, sistem akan melakukan proses perhitungan *Priority Scheduling* secara otomatis.



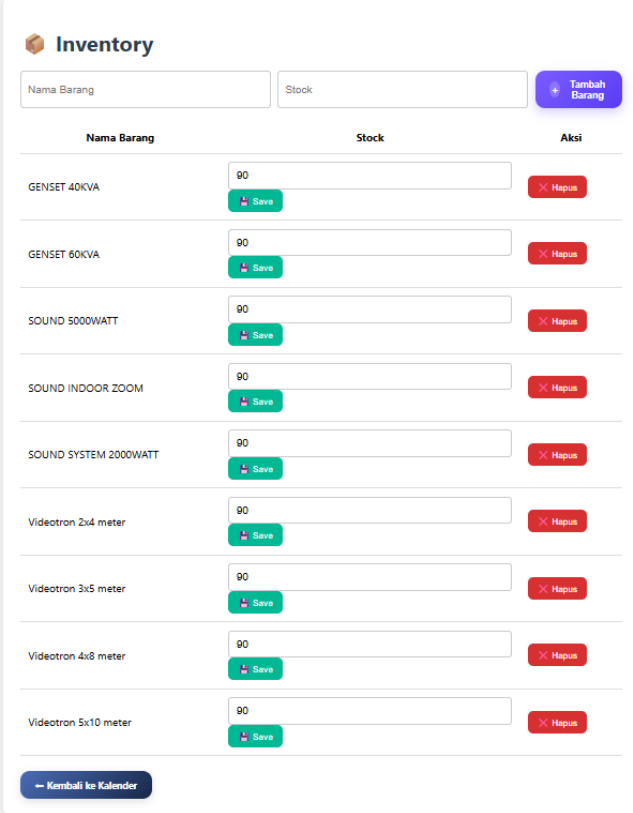

Gambar 4. Halaman *Input Event*

### 4.2.4 Halaman *Inventory*

Halaman *inventory* digunakan untuk mengelola data inventori barang *Event Organizer*. Pada halaman ini admin dapat:

1. menambahkan barang *inventori*,
2. mengubah jumlah stok barang,
3. menyimpan perubahan stok,
4. serta menghapus data barang.

Sistem akan menggunakan data inventori tersebut untuk proses pengecekan ketersediaan barang pada *event*.




Gambar 5. Halaman *Inventory*

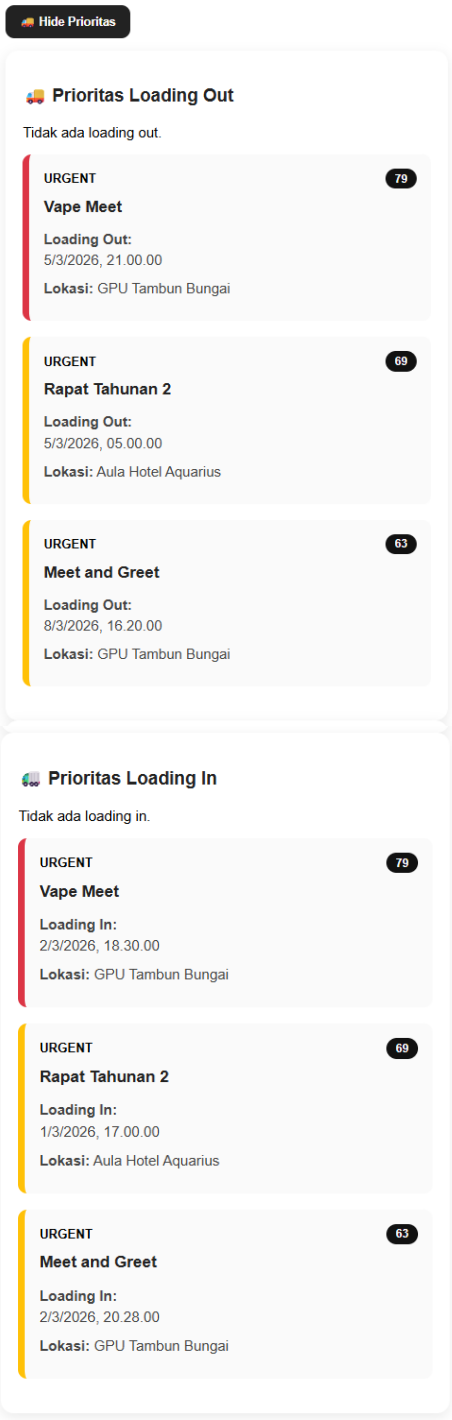
### 4.2.5 Halaman Prioritas *Event*

Halaman prioritas event digunakan untuk menampilkan hasil proses *Priority Scheduling* pada *Event Organizer*.

Pada halaman ini sistem menampilkan :

1. daftar prioritas *Loading In*,
2. daftar prioritas *Loading Out*,
3. nilai *Priority Score*,
4. lokasi *event*,
5. serta jadwal *Loading In* dan *Loading Out*.

Sistem akan melakukan *sorting event* berdasarkan nilai prioritas tertinggi sehingga *event* yang lebih penting akan diproses terlebih dahulu.




Gambar 6. Halaman Prioritas *Event*

### 4.2.6 Halaman Detail Event

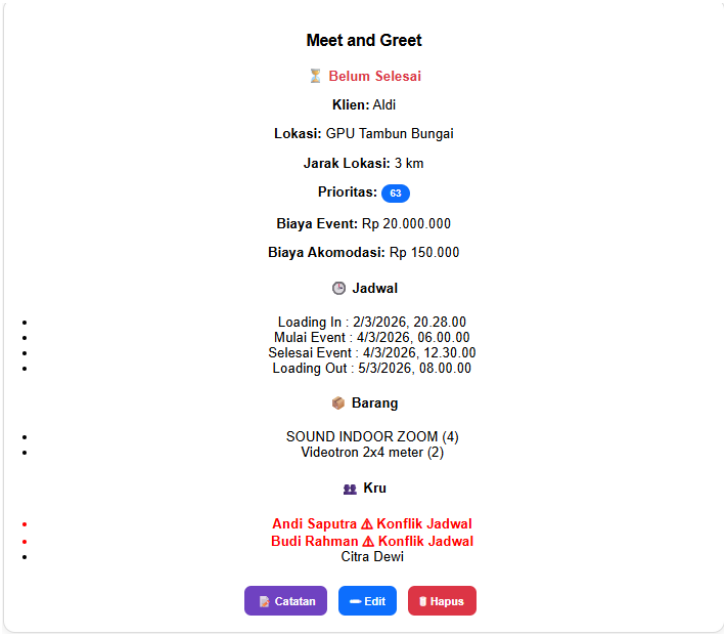
Halaman detail *event* digunakan untuk menampilkan informasi lengkap mengenai *event* yang telah dijadwalkan.

Informasi yang ditampilkan meliputi :

1. nama *event*,
2. status *event*,
3. nama klien,
4. lokasi *event*,
5. jarak lokasi,

6. *Priority Score*,
7. biaya *event*,
8. biaya akomodasi,
9. jadwal *event*,
10. barang yang digunakan,
11. serta kru yang bertugas.

Sistem juga menampilkan notifikasi konflik jadwal kru apabila terdapat kru yang digunakan pada waktu yang bersamaan.



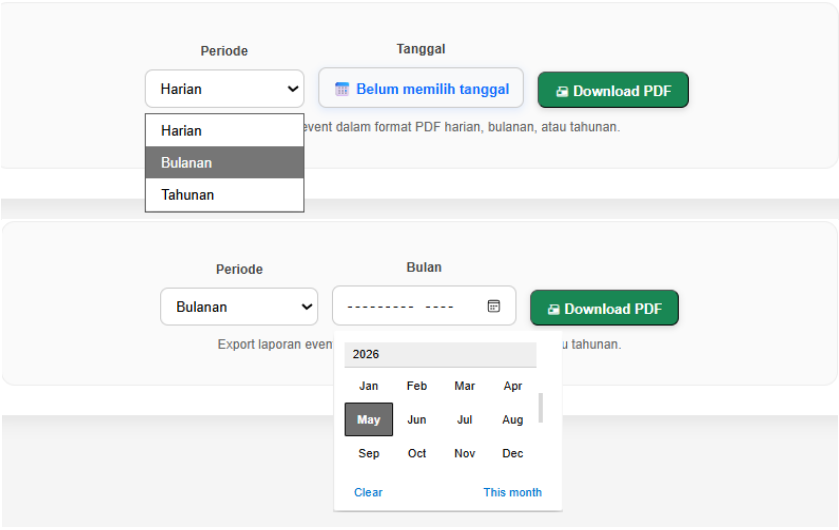

Gambar 7. Halaman Detail *Event*

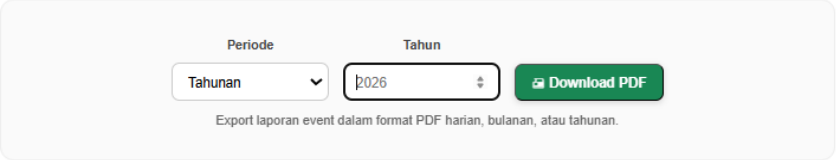
### 4.2.7 Halaman *Export* Laporan PDF

Halaman *Export* laporan digunakan untuk menghasilkan laporan *event* dalam format PDF. Sistem menyediakan beberapa pilihan periode laporan, yaitu :

1. harian,
2. bulanan,
3. dan tahunan.

Admin dapat memilih tanggal atau bulan tertentu sebelum melakukan proses *download* laporan PDF. Laporan yang dihasilkan berisi data *event* berdasarkan periode yang dipilih.






Gambar 8. Halaman *Export* Laporan PDF

### 4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fitur pada sistem dapat berjalan dengan baik sesuai kebutuhan pengguna. Metode pengujian yang digunakan pada penelitian ini adalah *Black Box Testing*, yaitu pengujian berdasarkan fungsi sistem tanpa melihat struktur kode program.

Pengujian dilakukan pada fitur utama sistem, seperti:

1. *login*,
2. pengelolaan *event*,
3. pengelolaan inventori,
4. proses *Scheduling*,
5. pengecekan konflik jadwal,
6. serta *Export* laporan PDF.

Tabel berikut menunjukkan hasil pengujian pada *fitur login* sistem.

Tabel 1. Pengujian Login

| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|---|---|
| 1 | *Username* dan *password* benar | Sistem berhasil *login* | Berhasil masuk ke *dashboard* | Valid |
| 2 | *Username* salah | Sistem menolak *login* | Muncul notifikasi gagal *login* | Valid |
| 3 | *Password* salah | Sistem menolak *login* | Muncul notifikasi gagal *login* | Valid |
| 4 | *Form* kosong | Sistem menolak *login* | Muncul validasi *form* | Valid |

Pengujian dilakukan untuk memastikan proses penambahan data *event* berjalan dengan baik.

Tabel 2. Pengujian *Input Event*

| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|---|---|
| 1 | Menambahkan data *event* lengkap | Data *event* berhasil disimpan | *Event* berhasil tampil pada sistem | Valid |
| 2 | *Form* data kosong | Sistem menolak penyimpanan | Muncul validasi *Input* | Valid |
| 3 | Menambahkan kru pada *event* | Kru tersimpan pada *event* | Data kru tampil pada detail *event* | Valid |
| 4 | Menambahkan barang *event* | Barang tersimpan pada *event* | Barang tampil pada detail *event* | Valid |

Pengujian dilakukan pada *fitur* pengelolaan *inventori* barang.

Tabel 3. Pengujian *Inventory*

| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|---|---|
| 1 | Menambahkan barang *inventori* | Barang berhasil ditambahkan | Data barang tampil pada *inventori* | Valid |
| 2 | Mengubah stok barang | Stok berhasil diperbarui | Data stok berubah | Valid |
| 3 | Menghapus barang | Barang berhasil dihapus | Barang tidak tampil pada *inventori* | Valid |

Pengujian dilakukan untuk memastikan sistem mampu melakukan *sorting* prioritas *event* secara otomatis.

Tabel 4. Pengujian *Priority Scheduling*

| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|---|---|
| 1 | Menghitung *Priority Score event* | Sistem menghasilkan nilai prioritas | *Priority Score* tampil pada *event* | Valid |
| 2 | Menampilkan prioritas *Loading In* | *Event* diurutkan berdasarkan prioritas | Daftar prioritas tampil | Valid |
| 3 | Menampilkan prioritas *Loading Out* | *Event* diurutkan berdasarkan prioritas | Daftar prioritas tampil | Valid |

Pengujian dilakukan untuk memastikan sistem mampu mendeteksi konflik penggunaan kru dan *inventori*.

Tabel 5. Pengujian Konflik Jadwal

| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|---|---|
| 1 | Kru digunakan pada dua *event* bersamaan | Sistem menampilkan konflik | Konflik jadwal tampil | Valid |
| 2 | Barang digunakan pada dua *event* bersamaan | Sistem menampilkan konflik | Konflik *inventori* tampil | Valid |

Pengujian dilakukan pada *fitur Export* laporan PDF.

Tabel 6. Pengujian *Export* PDF

| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian | Status |
|---|---|---|---|---|
| 1 | *Export* laporan harian | PDF berhasil dibuat | *File* PDF berhasil diunduh | Valid |
| 2 | *Export* laporan bulanan | PDF berhasil dibuat | *File* PDF berhasil diunduh | Valid |
| 3 | *Export* laporan tahunan | PDF berhasil dibuat | *File* PDF berhasil diunduh | Valid |

### 4.4 Analisis Hasil Implementasi Sistem

Berdasarkan hasil implementasi dan pengujian sistem, sistem penjadwalan *Loading In* dan *Loading Out* berbasis algoritma *Priority Scheduling* telah berhasil berjalan sesuai kebutuhan pengguna.

Sistem mampu:

1. melakukan proses penjadwalan *event* secara otomatis,
2. menghitung prioritas *event*,
3. mendeteksi konflik jadwal kru,

4. mendeteksi konflik penggunaan *inventori*,
5. serta membantu proses *Monitoring Event Organizer* secara lebih terstruktur.

Selain itu, penerapan algoritma *Priority Scheduling* juga membantu meningkatkan efisiensi proses penjadwalan dibandingkan metode manual. Hal tersebut dibuktikan melalui hasil pengukuran efisiensi waktu yang menunjukkan peningkatan efisiensi sebesar 40%.

Dengan adanya sistem ini, proses operasional *Event Organizer* menjadi lebih efektif, terstruktur, dan mudah dipantau oleh admin maupun pemilik *Event Organizer*.

## 5. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi sistem penjadwalan *Loading In* dan *Loading Out Event Organizer* berbasis algoritma *Priority Scheduling*, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Sistem penjadwalan berbasis *web* berhasil dibangun menggunakan *Framework Flask* dan *Database* MySQL untuk membantu proses pengelolaan *Event Organizer*.
2. Penerapan algoritma *Priority Scheduling* mampu membantu proses penentuan prioritas event berdasarkan parameter jarak lokasi, durasi *event*, biaya *event*, kompleksitas barang, dan jumlah kru.
3. Sistem mampu melakukan proses *Monitoring* jadwal *event, inventori* barang, serta kru secara lebih terstruktur dibandingkan metode manual.
4. Sistem berhasil mendeteksi konflik penggunaan kru dan *inventori* sehingga membantu admin dalam melakukan proses penyesuaian jadwal *event.*
5. Berdasarkan hasil pengujian, seluruh fitur utama pada sistem berjalan dengan baik dan sesuai kebutuhan pengguna.
6. Hasil pengukuran efisiensi menunjukkan bahwa sistem mampu meningkatkan efisiensi waktu proses penjadwalan sebesar 40% dibandingkan metode manual.

## 6. DAFTAR PUSTAKA

[1] Engineering ToolBox. (n.d.). Efficiency. Retrieved 2026, from https://www.engineeringtoolbox.com/efficiency-d_1768.html

[2] Grinberg, M. (2024). *The New and Improved Flask Mega-Tutorial*. Retrieved from https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world

[3] Lathkar, M. (2021). *Building Web Apps with Python and Flask.* BPB Publications. Retrieved from https://openlibrary.telkomuniversity.ac.id/home/catalog/id/178666/slug/building-web-apps-with-python-and-flask-learn-to-develop-and-deploy-responsive-restful-web-applications-using-flask-*Framework*.html

[4] Lucid *Software* Inc. (2024). *What is a UML Diagram?* Retrieved from https://www.lucidchart.com/pages/uml-diagram

[5] Rohmah, A. (2023). Implementasi Algoritma *Priority Scheduling* Sistem Informasi Pelayanan Administrasi Kependudukan Desa. Retrieved from https://ejournal.poltekharber.ac.id/index.php/informatika/article/view/4891/2477

[6] Wahid, A. A. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi.* Retrieved from https://www.researchgate.net/publication/346397070_Analisis_Metode_*Waterfall*_Untuk_Pengembangan_Sistem_Informasi

[7] Yanwari, M. I. (2023). *Priority Scheduling* Implementation for Exam Schedule. *Indonesian Journal of Information Systems, 5*(2), 80-89. doi:https://doi.org/10.24002/ijis.v5i2.6871

[8] Ye, S. K. (2023). *FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets.* Retrieved from https://arxiv.org/pdf/2307.10928