

Dokumentasi Sistem Multi-Agent Movie Booking

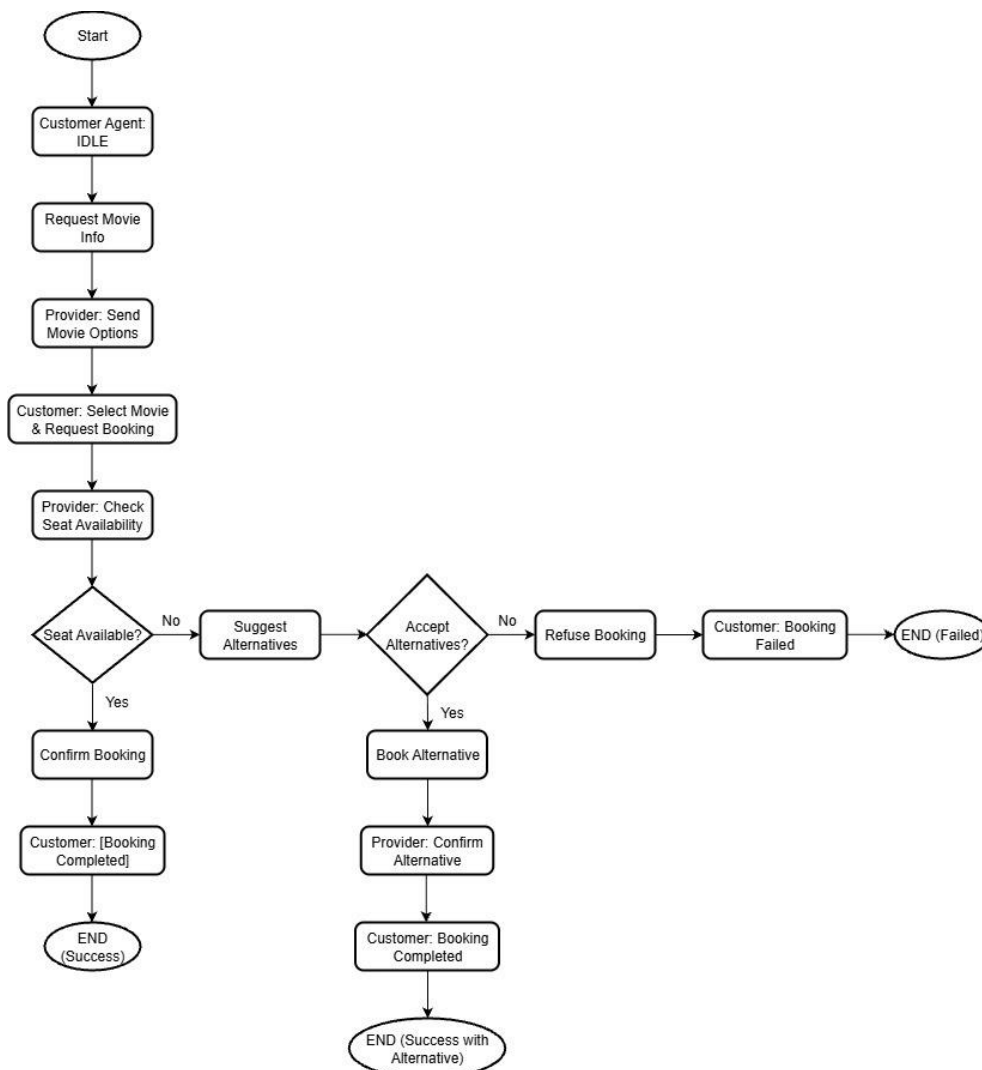
1. Skenario Terpilih dan Diagram Alur

1.1 Skenario Utama: Pemesanan Film dengan Negosiasi Kursi

Skenario Happy Path: 1. Customer meminta informasi film yang tersedia 2. Provider memberikan daftar film dan jadwal tayang 3. Customer memilih film dan meminta booking kursi spesifik 4. Provider memeriksa ketersediaan kursi 5. Provider mengkonfirmasi booking berhasil 6. Customer menerima konfirmasi dengan transaction ID

Skenario Error Path: 1. Customer meminta booking kursi yang sudah terpesan 2. Provider menolak dengan alasan kursi tidak tersedia 3. Provider menyarankan kursi alternatif 4. Customer memilih kursi alternatif 5. Provider mengkonfirmasi booking alternatif

1.2 Diagram Alur (Flowchart)



2. Protokol yang Dipakai dan Ontologi Pesan

2.1 Protokol FIPA-ACL

Sistem menggunakan protokol **FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language)** dengan performatif berikut:

Performatif	Deskripsi	Penggunaan
REQUEST	Permintaan informasi atau layanan	Customer → Provider (info film, booking)
INFORM	Memberikan informasi	Provider → Customer (daftar film, opsi kursi)
CONFIRM	Konfirmasi keberhasilan operasi	Provider → Customer (booking berhasil)
DISCONFIRM	Konfirmasi kegagalan operasi	Provider → Customer (kursi tidak tersedia)
AGREE	Persetujuan terhadap permintaan	Provider → Customer (setuju dengan booking)
REFUSE	Penolakan terhadap permintaan	Provider → Customer (tolak booking)
QUERY_IF	Query untuk mengecek ketersediaan	Customer → Provider (cek kursi)

2.2 Ontologi Pesan

MovieBookingOntology mendefinisikan struktur data:

Konsep Utama:

- **MOVIE_REQUEST**: Permintaan informasi film
- **BOOKING_REQUEST**: Permintaan booking kursi
- **BOOKING_RESPONSE**: Konfirmasi hasil booking
- **SEAT_AVAILABILITY**: Status ketersediaan kursi
- **ALTERNATIVE_REQUEST**: Permintaan kursi alternatif

Contoh Struktur Pesan:

REQUEST (Info Film):

Performative: REQUEST
Content: "REQUEST_INFO"
ConversationId: movie_booking_[timestamp]
ReplyWith: req_[timestamp]

INFORM (Daftar Film):

Performative: INFORM
Content: "Movie: Batman: The Dark Knight; Showtimes:"

10:00,13:00,16:00,19:00,22:00"
ConversationId: movie_booking_[timestamp]
InReplyTo: req_[timestamp]

CONFIRM (Booking Berhasil):

Performative: CONFIRM
Content: "Booking berhasil! Transaction ID: TXN_[timestamp]; Kursi: A1,A2;
Waktu: 19:00"
ConversationId: movie_booking_[timestamp]
InReplyTo: booking_[timestamp]

3. Rekaman Log

3.1 Happy Path Log

Contoh Log CSV (Happy Path):

```
timestamp,sender,receiver,performative,conversationId,content,level
2025-09-04 17:43:08,SYSTEM,SYSTEM,SYSTEM_STARTUP,system,Multi-Agent Booking
System started,INFO
2025-09-04
17:43:22,customer,provider,REQUEST,movie_booking_1,BOOKING:Time=10:00;Seats=A
1;Class=VIP,INFO
2025-09-04 17:43:22,provider,customer,CONFIRM,movie_booking_1,Booking
berhasil! Transaction ID: TXN_1,INFO
2025-09-04 17:43:22,customer,SYSTEM,STATE_CHANGE,movie_booking_1,State
changed to BOOKING_COMPLETED,INFO
```

3.2 Error Path Log

Contoh Log CSV (Error Path):

```
timestamp,sender,receiver,performative,conversationId,content,level
2025-09-04
17:44:15,customer,provider,REQUEST,movie_booking_2,BOOKING:Time=19:00;Seats=A
1;Class=VIP,INFO
2025-09-04 17:44:15,provider,customer,DISCONFIRM,movie_booking_2,Kursi A1
tidak tersedia,INFO
2025-09-04 17:44:15,provider,customer,INFORM,movie_booking_2,Alternatif:
A2,A3 tersedia,INFO
2025-09-04
17:44:16,customer,provider,REQUEST,movie_booking_2,BOOKING:Time=19:00;Seats=A
2;Class=VIP,INFO
2025-09-04 17:44:16,provider,customer,CONFIRM,movie_booking_2,Booking
berhasil! Transaction ID: TXN_2,INFO
```

4. Analisis: Keputusan Desain

4.1 State Management

Keputusan Desain State:

```
enum AgentState {  
    IDLE, // Agen dalam keadaan standby  
    REQUESTING_INFO, // Meminta informasi film  
    RECEIVED_OPTIONS, // Menerima opsi film  
    REQUESTING_BOOKING, // Meminta booking kursi  
    WAITING_CONFIRMATION, // Menunggu konfirmasi  
    BOOKING_COMPLETED, // Booking selesai  
    ERROR, // Terjadi error  
    CHECKING_SEAT // Mengecek ketersediaan kursi  
}
```

Alasan Keputusan: - **State-based approach** memudahkan tracking status agen - **Thread-safe state management** dengan ReentrantLock - **Atomic operations** untuk counter dan state changes - **Timeout handling** untuk mencegah deadlock

4.2 Penanganan Error

Strategi Error Handling:

1. Provider menyarankan kursi alternatif saat kursi tidak tersedia
2. Customer dapat memilih alternatif atau membatalkan

4.3 Alternatif Desain yang Dipertimbangkan

1. Synchronous vs Asynchronous Communication: - **Dipilih:** Asynchronous dengan callback - **Alasan:** Non-blocking, scalable, real-time updates

2. REST API vs FIPA-ACL: - **Dipilih:** FIPA-ACL - **Alasan:** Standard multi-agent, rich semantics, interoperability

4.4 Performance Considerations

Optimisasi yang Diterapkan: - **Connection pooling** untuk agent communication - **Lazy loading** untuk seat information - **Efficient locking strategy** (read-write locks) - **Periodic cleanup** untuk old logs

5. Kesimpulan

Sistem Multi-Agent Movie Booking ini mengimplementasikan:

1. **Protokol FIPA-ACL** yang standar dan interoperable
2. **State management** yang robust dengan error handling
3. **Alternative negotiation** untuk user experience yang baik
4. **Comprehensive logging** untuk monitoring dan debugging