

BAB III

PERANCANGAN RINCI

Pengujian Unit

Algoritma Register Pelanggan (ALGO_BALLS_001)

Nama Kelas: Authentication

Nama Operasi: registerUser()

Tabel 3.1 Algoritma Register Pelanggan

...

```
Public function registerUser() {  
  IF (submit)  
    THEN  
      IF(registerUser)  
        THEN  
          value = Read(username, email, password, confirmPassword, fullName, phoneNumber)  
          IF(value != null AND password == confirmPassword)  
            THEN  
              existingUser = prisma.user.findFirst({  
                where: { OR: [{ username }, { email }] }  
              })  
              IF(existingUser == null)  
                THEN  
                  hashedPassword = bcrypt.hash(password, 10)  
                  newUser = prisma.user.create({  
                    data: { username, email, password: hashedPassword, fullName, phoneNumber, role: 'CUSTOMER' }  
                  })  
                  print 'Registration successful'  
                  redirect('/login')  
                ELSE  
                  THEN  
                    print('Username or email already exists')  
                    redirect('/register')  
                  ENDIF  
                ELSE  
                  THEN  
                    print('Invalid data or passwords do not match')  
                    redirect('/register')  
                  ENDIF  
                ELSE  
                  THEN
```

```

print 'Registration data incomplete'
load view('/register')
ENDIF
ELSE
THEN
print 'Form not submitted'
load view('register')
ENDIF
}
...

```

Flow graph:

Gambar 3.1 Flow Graph Register Pelanggan

Jalur independen:

- 1). 1 – 9 - 10
- 2). 1 – 2 – 7 – 8 – 10
- 3). 1 - 2 - 3 - 6 - 8 - 10
- 4). 1 - 2 - 3 - 4 - 6 - 8 - 10

Perhitungan Cyclomatic Complexity / V(G):

- $V(G) = 4$ Region
- $V(G) = E - N + 2 = 12 - 10 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

Test Case & Hasil:

Tabel 3.2 Test case dan hasil algoritma Register Pelanggan

No	No. Jalur	Data Input	Expected Result	Result	Status
1	1	Form register tidak tersubmit	Menampilkan halaman register secara tetap	Menampilkan halaman register secara tetap	Valid
2	2	Form register tidak terisi lengkap	Menampilkan pesan error "All fields are required"	Menampilkan pesan error "All fields are required"	Valid
3	3	Form register terisi lengkap tetapi username/email sudah ada	Menampilkan pesan error "Username or email already exists"	Menampilkan pesan error "Username or email already exists"	Valid
4	4	Form register terisi lengkap, data valid dan tersubmit	Menyimpan data user ke database & mengalihkan ke halaman login	Menyimpan data user ke database & mengalihkan ke halaman login	Valid

Algoritma Login Akun (ALGO_BALLS_002)

Nama Kelas: Authentication

Nama Operasi: loginUser()

Tabel 3.3 Algoritma Login Akun

...

```
Public function loginUser() {  
  IF (submit)  
  THEN  
    IF(loginUser)  
    THEN  
      value = Read(emailOrUsername, password)  
      IF(value != null)  
      THEN  
        user = prisma.user.findFirst({  
          where: { OR: [{ email: emailOrUsername }, { username: emailOrUsername }] }  
        })  
        IF(user != null)  
        THEN  
          isPasswordValid = bcrypt.compare(password, user.password)  
          IF(isPasswordValid)  
          THEN  
            token = jwt.sign({ id: user.id, role: user.role }, JWT_SECRET)  
            cookies.set('auth-token', token)  
            print 'Login successful'  
            IF(user.role == 'CUSTOMER')  
            redirect('/pengguna')  
            ELSE IF(user.role == 'ADMIN')  
            redirect('/admin')  
            ELSE IF(user.role == 'SUPER_ADMIN')  
            redirect('/super-admin')  
          ENDIF  
        ELSE  
          print('Invalid credentials')  
          redirect('/login')  
        ENDIF  
      ELSE  
        print('Invalid credentials')  
        redirect('/login')  
      ENDIF  
    ELSE  
      THEN  
        print('Email/username and password are required')  
        redirect('/login')
```

```

ENDIF
ELSE
THEN
print 'Login data incomplete'
load view('/login')
ENDIF
ELSE
THEN
print 'Form not submitted'
load view('login')
ENDIF
}
...

```

Flow graph:

Gambar 3.2 Flow Graph Login Akun

Jalur independen:

- 1). 1 – 11 - 12
- 2). 1 – 2 – 9 – 10 – 12
- 3). 1 - 2 - 3 - 8 - 10 - 12
- 4). 1 - 2 - 3 - 4 - 7 - 10 - 12
- 5). 1 - 2 - 3 - 4 - 5 - 6 - 10 - 12

Perhitungan Cyclomatic Complexity / $V(G)$:

- $V(G) = 5$ Region
- $V(G) = E - N + 2 = 15 - 12 + 2 = 5$
- $V(G) = P + 1 = 4 + 1 = 5$

Test Case & Hasil:

Tabel 3.4 Test case dan hasil algoritma Login Akun

No	No. Jalur	Data Input	Expected Result	Result	Status
1	1	Form login tidak tersubmit	Menampilkan halaman login secara tetap	Menampilkan halaman login secara tetap	Valid
2	2	Form login tidak terisi lengkap	Menampilkan pesan error "Email/username and password are required"	Menampilkan pesan error "Email/username and password are required"	Valid

No	No. Jalur	Data Input	Expected Result	Result	Status
3	3	Form login terisi lengkap tetapi user tidak ditemukan	Menampilkan pesan error "Invalid credentials"	Menampilkan pesan error "Invalid credentials"	Valid
4	4	Form login terisi lengkap tetapi password salah	Menampilkan pesan error "Invalid credentials"	Menampilkan pesan error "Invalid credentials"	Valid
5	5	Form login terisi lengkap, kredensial valid dan tersubmit	Login berhasil & mengalihkan ke dashboard sesuai role	Login berhasil & mengalihkan ke dashboard sesuai role	Valid

Algoritma Booking Lapangan (ALGO_BALLS_003)

Nama Kelas: Booking

Nama Operasi: createBooking()

Tabel 3.5 Algoritma Booking Lapangan

...

```

Public function createBooking() {
  IF (submit)
  THEN
    IF(createBooking)
    THEN
      value = Read(fieldId, bookingDate, startTime, endTime)
      IF(value != null AND startTime < endTime)
      THEN
        conflictBooking = prisma.booking.findFirst({
          where: {
            fieldId: fieldId,
            bookingDate: bookingDate,
            OR: [
              { startTime: { lte: startTime }, endTime: { gt: startTime } },
              { startTime: { lt: endTime }, endTime: { gte: endTime } }
            ]
          }
        })
        IF(conflictBooking == null)
        THEN
          duration = calculateDuration(startTime, endTime)
          field = prisma.field.findUnique({ where: { id: fieldId } })
          amount = duration * field.hourlyRate

```

```

booking = prisma.booking.create({
data: { userId, fieldId, bookingDate, startTime, endTime, duration, amount, status: 'UNPAID' }
})
print 'Booking created successfully'
redirect('/pengguna/booking')
ELSE
THEN
print('Time slot not available')
redirect('/pengguna/booking')
ENDIF
ELSE
THEN
print('Invalid booking data')
redirect('/pengguna/booking')
ENDIF
ELSE
THEN
print 'Booking data incomplete'
load view('/pengguna/booking')
ENDIF
ELSE
THEN
print 'Form not submitted'
load view('booking')
ENDIF
}
...

```

Flow graph:

Gambar 3.3 Flow Graph Booking Lapangan

PROF

Jalur independen:

- 1). 1 – 9 - 10
- 2). 1 – 2 – 7 – 8 – 10
- 3). 1 - 2 - 3 - 6 - 8 - 10
- 4). 1 - 2 - 3 - 4 - 6 - 8 - 10

Perhitungan Cyclomatic Complexity / $V(G)$:

- $V(G) = 4$ Region
- $V(G) = E - N + 2 = 12 - 10 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

Test Case & Hasil:

Tabel 3.6 Test case dan hasil algoritma Booking Lapangan

No	No. Jalur	Data Input	Expected Result	Result	Status
1	1	Form booking tidak tersubmit	Menampilkan halaman booking secara tetap	Menampilkan halaman booking secara tetap	Valid
2	2	Form booking tidak terisi lengkap	Menampilkan pesan error "Booking data incomplete"	Menampilkan pesan error "Booking data incomplete"	Valid
3	3	Form booking terisi lengkap tetapi waktu konflik	Menampilkan pesan error "Time slot not available"	Menampilkan pesan error "Time slot not available"	Valid
4	4	Form booking terisi lengkap, data valid dan tersubmit	Menyimpan booking ke database & mengalihkan ke halaman booking	Menyimpan booking ke database & mengalihkan ke halaman booking	Valid

Algoritma Upload Bukti Pembayaran (ALGO_BALLS_004)

Nama Kelas: Payment

Nama Operasi: uploadPaymentProof()

Tabel 3.7 Algoritma Upload Bukti Pembayaran

...

```

Public function uploadPaymentProof() {
  IF (submit)
  THEN
    IF(uploadPaymentProof)
  THEN
    value = Read(bookingId, paymentMethod, proofImage)
    IF(value != null AND proofImage.type.includes('image'))
  THEN
    booking = prisma.booking.findUnique({ where: { id: bookingId } })
    IF(booking != null AND booking.status == 'UNPAID')
  THEN
    fileName = generateFileName(proofImage)
    optimizedImage = sharp(proofImage).webp().toBuffer()
    saveFile(fileName, optimizedImage)
    payment = prisma.payment.create({
      data: { bookingId, paymentMethod, proofImage: fileName, status: 'PENDING' }
    })
    prisma.booking.update({
      where: { id: bookingId },
      data: { status: 'PENDING' }
    })
  }
}

```

```

})
print 'Payment proof uploaded successfully'
redirect('/pengguna/transactions')
ELSE
THEN
print('Invalid booking or booking already paid')
redirect('/pengguna/transactions')
ENDIF
ELSE
THEN
print('Invalid file or payment data')
redirect('/pengguna/transactions')
ENDIF
ELSE
THEN
print 'Payment data incomplete'
load view('/pengguna/transactions')
ENDIF
ELSE
THEN
print 'Form not submitted'
load view('upload-payment')
ENDIF
}
...

```

Flow graph:

Gambar 3.4 Flow Graph Upload Bukti Pembayaran

Jalur independen:

- 1). 1 – 9 - 10
- 2). 1 – 2 – 7 – 8 – 10
- 3). 1 - 2 - 3 - 6 - 8 - 10
- 4). 1 - 2 - 3 - 4 - 6 - 8 - 10

Perhitungan Cyclomatic Complexity / $V(G)$:

- $V(G) = 4$ Region
- $V(G) = E - N + 2 = 12 - 10 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

Test Case & Hasil:

Tabel 3.8 Test case dan hasil algoritma Upload Bukti Pembayaran

No	No. Jalur	Data Input	Expected Result	Result	Status
----	--------------	------------	-----------------	--------	--------

No	No. Jalur	Data Input	Expected Result	Result	Status
1	1	Form upload tidak tersubmit	Menampilkan halaman upload secara tetap	Menampilkan halaman upload secara tetap	Valid
2	2	Form upload tidak terisi lengkap	Menampilkan pesan error "Payment data incomplete"	Menampilkan pesan error "Payment data incomplete"	Valid
3	3	Form upload terisi lengkap tetapi booking tidak valid	Menampilkan pesan error "Invalid booking or booking already paid"	Menampilkan pesan error "Invalid booking or booking already paid"	Valid
4	4	Form upload terisi lengkap, data valid dan tersubmit	Menyimpan bukti pembayaran & mengalihkan ke halaman transaksi	Menyimpan bukti pembayaran & mengalihkan ke halaman transaksi	Valid

Algoritma CRUD Program Loyalty (ALGO_BALLS_005)

Nama Kelas: LoyaltyProgram

Nama Operasi: createReward()

Tabel 3.9 Algoritma CRUD Program Loyalty

...

```

Public function createReward() {
  IF (submit)
  THEN
    IF(createReward AND isAdmin)
    THEN
      value = Read(name, description, pointsRequired, image)
      IF(value != null AND pointsRequired > 0)
      THEN
        reward = prisma.loyaltyProgram.create({
          data: { name, description, pointsRequired, image, isActive: true }
        })
        print 'Reward created successfully'
        redirect('/admin/loyalty')
      ELSE
      THEN
        print('Invalid reward data')
        redirect('/admin/loyalty')
      ENDIF
    ELSE
    THEN
      print 'Unauthorized or data incomplete'
    
```

```

load view('/admin/loyalty')
ENDIF
ELSE
THEN
print 'Form not submitted'
load view('create-reward')
ENDIF
}
...

```

Flow graph:

Gambar 3.5 Flow Graph CRUD Program Loyalty

Jalur independen:

- 1). 1 – 7 - 8
- 2). 1 – 2 – 5 – 6 – 8
- 3). 1 - 2 - 3 - 5 - 6 - 8
- 4). 1 - 2 - 3 - 4 - 6 - 8

Perhitungan Cyclomatic Complexity / $V(G)$:

- $V(G) = 4$ Region
- $V(G) = E - N + 2 = 10 - 8 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

Test Case & Hasil:

Tabel 3.10 Test case dan hasil algoritma CRUD Program Loyalty

No	No. Jalur	Data Input	Expected Result	Result	Status
1	1	Form reward tidak tersubmit	Menampilkan halaman create reward secara tetap	Menampilkan halaman create reward secara tetap	Valid
2	2	Form reward tidak terisi lengkap atau user bukan admin	Menampilkan pesan error "Unauthorized or data incomplete"	Menampilkan pesan error "Unauthorized or data incomplete"	Valid
3	3	Form reward terisi lengkap tetapi data tidak valid	Menampilkan pesan error "Invalid reward data"	Menampilkan pesan error "Invalid reward data"	Valid
4	4	Form reward terisi lengkap, data valid dan tersubmit	Menyimpan reward ke database & mengalihkan ke halaman loyalty	Menyimpan reward ke database & mengalihkan ke halaman loyalty	Valid

Algoritma Penukaran Loyalty Poin oleh Pelanggan (ALGO_BALLS_006)

Nama Kelas: LoyaltyRedemption

Nama Operasi: redeemPoints()

Tabel 3.11 Algoritma Penukaran Loyalty Poin oleh Pelanggan

...

```
Public function redeemPoints() {
  IF (submit)
  THEN
    IF(redeemPoints)
    THEN
      value = Read(userId, rewardId)
      IF(value != null)
      THEN
        userPoints = prisma.userPoint.findFirst({
          where: { userId: userId, isActive: true }
        })
        reward = prisma.loyaltyProgram.findUnique({
          where: { id: rewardId }
        })
        IF(userPoints != null AND reward != null)
        THEN
          IF(userPoints.points >= reward.pointsRequired)
          THEN
            newPoints = userPoints.points - reward.pointsRequired
            prisma.userPoint.update({
              where: { id: userPoints.id },
              data: { points: newPoints }
            })
            qrCode = generateQRCode(userId, rewardId)
            redemption = prisma.redemption.create({
              data: { userId, rewardId, pointsUsed: reward.pointsRequired, qrCode, status: 'PENDING' }
            })
            print 'Points redeemed successfully'
            redirect('/pengguna/loyalty')
          ELSE
            THEN
              print('Insufficient points for this reward')
              redirect('/pengguna/loyalty')
            ENDIF
          ELSE
            THEN
              print('Invalid user points or reward not found')
              redirect('/pengguna/loyalty')
            ENDIF
          ELSE
            THEN
```

```

print('Invalid redemption data')
redirect('/pengguna/loyalty')
ENDIF
ELSE
THEN
print 'Redemption data incomplete'
load view('/pengguna/loyalty')
ENDIF
ELSE
THEN
print 'Form not submitted'
load view('redeem-reward')
ENDIF
}
...

```

Flow graph:

Gambar 3.6 Flow Graph Penukaran Loyalty Poin oleh Pelanggan

Jalur independen:

- 1). 1 – 9 - 10
- 2). 1 – 2 – 7 – 8 – 10
- 3). 1 - 2 - 3 - 6 - 8 - 10
- 4). 1 - 2 - 3 - 4 - 5 - 8 - 10
- 5). 1 - 2 - 3 - 4 - 5 - 6 - 8 - 10

Perhitungan Cyclomatic Complexity / V(G):

- $V(G) = 5$ Region
- $V(G) = E - N + 2 = 13 - 10 + 2 = 5$
- $V(G) = P + 1 = 4 + 1 = 5$

PROF

Test Case & Hasil:

Tabel 3.12 Test case dan hasil algoritma Penukaran Loyalty Poin oleh Pelanggan

No	No. Jalur	Data Input	Expected Result	Result	Status
1	1	Form redeem tidak tersubmit	Menampilkan halaman redeem reward secara tetap	Menampilkan halaman redeem reward secara tetap	Valid
2	2	Form redeem tidak terisi lengkap	Menampilkan pesan error "Redemption data incomplete"	Menampilkan pesan error "Redemption data incomplete"	Valid

No	No. Jalur	Data Input	Expected Result	Result	Status
3	3	Form redeem terisi lengkap tetapi data tidak valid	Menampilkan pesan error "Invalid redemption data"	Menampilkan pesan error "Invalid redemption data"	Valid
4	4	Form redeem terisi lengkap tetapi user/reward tidak ditemukan	Menampilkan pesan error "Invalid user points or reward not found"	Menampilkan pesan error "Invalid user points or reward not found"	Valid
5	5	Form redeem terisi lengkap tetapi poin tidak mencukupi	Menampilkan pesan error "Insufficient points for this reward"	Menampilkan pesan error "Insufficient points for this reward"	Valid
6	6	Form redeem terisi lengkap, poin mencukupi dan tersubmit	Mengurangi poin user, generate QR code & mengalihkan ke halaman loyalty	Mengurangi poin user, generate QR code & mengalihkan ke halaman loyalty	Valid

Pengujian Validasi

Requirement Register Pelanggan (SRS_F-BALLS-001)

Pengujian Register Pelanggan (PV_BALLS_01)

Test Case: Semua Field Diisi

Prosedur:

1. Pengguna menuju halaman "Register"
2. Pengguna mengisi semua field form register pada halaman tersebut
3. Pengguna mengklik button "Sign Up" pada form register

Expected Result:

Sistem akan menyimpan data ke dalam database dan sistem akan mengalihkan ke halaman login dengan pesan sukses registrasi.

Requirement Register Pelanggan (SRS_F-BALLS-001)

Pengujian Register Pelanggan (PV_BALLS_02)

Test Case: Field Diisi Tidak Lengkap

Prosedur:

1. Pengguna menuju halaman "Register"
2. Pengguna mengisi field form register pada halaman tersebut tetapi tidak lengkap
3. Pengguna mengklik button "Sign Up" pada form register

Expected Result:

Sistem akan menampilkan pesan error "All fields are required" pada field yang belum terisi.

Requirement Login Akun (SRS_F-BALLS-002)

Pengujian Login Akun (PV_BALLS_03)

Test Case: Email/Username dan Password Valid

Prosedur:

1. Pengguna menuju halaman "Login"
2. Pengguna mengisi field email/username dan password pada halaman tersebut
3. Pengguna mengklik button "Sign In" pada halaman tersebut

Expected Result:

Sistem akan mengecek data email/username dan password ke database, jika datanya valid, maka sistem akan mengalihkan ke halaman dashboard sesuai role pengguna.

Requirement Login Akun (SRS_F-BALLS-002)

Pengujian Login Akun (PV_BALLS_04)

Test Case: Email/Username dan Password Tidak Valid

Prosedur:

1. Pengguna menuju halaman "Login"
2. Pengguna mengisi field email/username dan password pada halaman tersebut tetapi tidak valid
3. Pengguna mengklik button "Sign In" pada halaman tersebut

Expected Result:

Sistem akan mengecek data email/username dan password ke database, jika datanya tidak valid, maka sistem akan menampilkan pesan error "Invalid credentials".

PROF

Requirement Booking Lapangan (SRS_F-BALLS-003)

Pengujian Booking Lapangan (PV_BALLS_05)

Test Case: Semua Field Diisi

Prosedur:

1. Pelanggan menuju halaman "Booking"
2. Pelanggan memilih lapangan yang diinginkan
3. Pelanggan mengisi tanggal, waktu mulai, dan waktu selesai
4. Pelanggan mengklik button "Create Booking"

Expected Result:

Sistem akan menyimpan data booking ke dalam database dan sistem akan menampilkan pesan bahwa booking berhasil dibuat dengan status UNPAID.

Requirement Booking Lapangan (SRS_F-BALLS-003)

Pengujian Booking Lapangan (PV_BALLS_06)

Test Case: Waktu Konflik dengan Booking Lain

Prosedur:

1. Pelanggan menuju halaman "Booking"
2. Pelanggan memilih lapangan yang sudah diboeking pada waktu tertentu
3. Pelanggan mengisi tanggal dan waktu yang sama dengan booking yang sudah ada
4. Pelanggan mengklik button "Create Booking"

Expected Result:

Sistem akan menampilkan pesan error "Time slot not available" karena terjadi konflik waktu.

Requirement Upload Bukti Pembayaran (SRS_F-BALLS-004)

Pengujian Upload Bukti Pembayaran (PV_BALLS_07)

Test Case: File Gambar Valid

Prosedur:

1. Pelanggan menuju halaman "Upload Payment Proof"
2. Pelanggan mengisi Booking ID yang valid
3. Pelanggan memilih metode pembayaran
4. Pelanggan mengupload file gambar bukti pembayaran
5. Pelanggan mengklik button "Upload"

Expected Result:

Sistem akan menyimpan bukti pembayaran ke database dan mengubah status booking menjadi PENDING untuk menunggu verifikasi admin.

PROF

Requirement Upload Bukti Pembayaran (SRS_F-BALLS-004)

Pengujian Upload Bukti Pembayaran (PV_BALLS_08)

Test Case: File Bukan Gambar

Prosedur:

1. Pelanggan menuju halaman "Upload Payment Proof"
2. Pelanggan mengisi Booking ID yang valid
3. Pelanggan memilih metode pembayaran
4. Pelanggan mengupload file yang bukan gambar (PDF, DOC, dll)
5. Pelanggan mengklik button "Upload"

Expected Result:

Sistem akan menampilkan pesan error "Invalid file type. Please upload an image file" karena file yang

Requirement Melihat Riwayat Booking (SRS_F-BALLS-005)

Pengujian Melihat Riwayat Booking (PV_BALLS_09)

Test Case: Pelanggan Memiliki Booking

Prosedur:

1. Pelanggan login ke sistem
2. Pelanggan menuju halaman "Transactions" atau "Booking History"
3. Sistem menampilkan daftar booking pelanggan

Expected Result:

Sistem akan menampilkan semua riwayat booking pelanggan dengan informasi lengkap seperti tanggal, waktu, lapangan, status, dan total pembayaran.

Requirement Melihat Poin Loyalty (SRS_F-BALLS-006)

Pengujian Melihat Poin Loyalty (PV_BALLS_10)

Test Case: Pelanggan Memiliki Poin

Prosedur:

1. Pelanggan login ke sistem
2. Pelanggan menuju halaman "Loyalty"
3. Sistem menampilkan total poin yang dimiliki pelanggan

Expected Result:

Sistem akan menampilkan total poin loyalty pelanggan dan riwayat perolehan poin dari booking yang telah completed.

Requirement Tukar Poin Loyalty (SRS_F-BALLS-007)

Pengujian Tukar Poin Loyalty (PV_BALLS_11)

Test Case: Poin Mencukupi untuk Reward

Prosedur:

1. Pelanggan menuju halaman "Loyalty"
2. Pelanggan memilih reward yang diinginkan
3. Pelanggan mengklik button "Redeem" pada reward tersebut
4. Sistem mengecek apakah poin pelanggan mencukupi

Expected Result:

Sistem akan mengurangi poin pelanggan sesuai requirement reward dan memberikan QR code untuk klaim reward.

Requirement Tukar Poin Loyalty (SRS_F-BALLS-007)

Pengujian Tukar Poin Loyalty (PV_BALLS_12)

Test Case: Poin Tidak Mencukupi untuk Reward

Prosedur:

1. Pelanggan menuju halaman "Loyalty"
2. Pelanggan memilih reward yang membutuhkan poin lebih banyak dari yang dimiliki
3. Pelanggan mengklik button "Redeem" pada reward tersebut

Expected Result:

Sistem akan menampilkan pesan error "Insufficient points. You need X more points to redeem this reward".

Requirement CRUD Booking oleh Admin (SRS_F-BALLS-009)

Pengujian CRUD Booking oleh Admin (PV_BALLS_13)

Test Case: Admin Mengubah Status Booking

Prosedur:

1. Admin login ke sistem
2. Admin menuju halaman "Bookings"
3. Admin memilih booking yang akan diubah statusnya
4. Admin mengubah status booking (PENDING → CONFIRMED)
5. Admin mengklik button "Update Status"

Expected Result:

Sistem akan mengubah status booking sesuai pilihan admin dan menyimpan perubahan ke database.

Requirement CRUD Program Loyalty oleh Admin (SRS_F-BALLS-011)

Pengujian CRUD Program Loyalty oleh Admin (PV_BALLS_14)

Test Case: Admin Menambah Reward Baru

Prosedur:

1. Admin login ke sistem
2. Admin menuju halaman "Loyalty Program"
3. Admin mengklik button "Add New Reward"
4. Admin mengisi semua field reward (nama, deskripsi, poin required, gambar)
5. Admin mengklik button "Create Reward"

Expected Result:

Sistem akan menyimpan reward baru ke database dan menampilkan reward tersebut di katalog loyalty program.

Requirement CRUD Program Loyalty oleh Admin (SRS_F-BALLS-011)

Pengujian CRUD Program Loyalty oleh Admin (PV_BALLS_15)

Test Case: Admin Menambah Reward dengan Field Kosong

Prosedur:

1. Admin login ke sistem
2. Admin menuju halaman "Loyalty Program"
3. Admin mengklik button "Add New Reward"
4. Admin mengisi field reward tetapi tidak lengkap
5. Admin mengklik button "Create Reward"

Expected Result:

Sistem akan menampilkan pesan error "Please fill out this field!" pada field yang belum terisi.

Requirement Laporan Transaksi (SRS_F-BALLS-015)

Pengujian Laporan Transaksi (PV_BALLS_16)

Test Case: Admin Melihat Laporan dengan Data Tersedia

Prosedur:

1. Admin login ke sistem
2. Admin menuju halaman "Reports"
3. Admin memilih periode laporan (tanggal mulai dan tanggal akhir)
4. Admin mengklik button "Generate Report"

Expected Result:

Sistem akan menampilkan laporan transaksi dengan data statistik seperti total booking, total revenue, dan grafik tren booking.

Requirement Verifikasi Transaksi (SRS_F-BALLS-016)

Pengujian Verifikasi Transaksi (PV_BALLS_17)

Test Case: Admin Memverifikasi Pembayaran Valid

Prosedur:

1. Admin login ke sistem
2. Admin menuju halaman "Transactions"
3. Admin memilih pembayaran dengan status PENDING
4. Admin melihat bukti pembayaran yang diupload
5. Admin mengklik button "Confirm Payment"

Expected Result:

Sistem akan mengubah status pembayaran menjadi PAID dan status booking menjadi CONFIRMED.

Requirement Verifikasi Transaksi (SRS_F-BALLS-016)

Pengujian Verifikasi Transaksi (PV_BALLS_18)

Test Case: Admin Menolak Pembayaran Tidak Valid

Prosedur:

1. Admin login ke sistem
2. Admin menuju halaman "Transactions"
3. Admin memilih pembayaran dengan status PENDING
4. Admin melihat bukti pembayaran yang tidak valid
5. Admin mengklik button "Reject Payment" dan memberikan catatan

Expected Result:

Sistem akan mengubah status pembayaran menjadi INVALID dan memberikan notifikasi ke pelanggan untuk mengupload ulang bukti pembayaran.

Requirement CRUD Admin oleh Super Admin (SRS_F-BALLS-017)

Pengujian CRUD Admin oleh Super Admin (PV_BALLS_19)

Test Case: Super Admin Menambah Admin Baru

Prosedur:

1. Super Admin login ke sistem
2. Super Admin menuju halaman "Admin Management"
3. Super Admin mengklik button "Add New Admin"
4. Super Admin mengisi semua field admin (username, email, password, full name)
5. Super Admin mengklik button "Create Admin"

Expected Result:

Sistem akan membuat akun admin baru dengan role ADMIN dan menyimpan data ke database.

PROF

Requirement CRUD Admin oleh Super Admin (SRS_F-BALLS-017)

Pengujian CRUD Admin oleh Super Admin (PV_BALLS_20)

Test Case: Super Admin Menambah Admin dengan Field Kosong

Prosedur:

1. Super Admin login ke sistem
2. Super Admin menuju halaman "Admin Management"
3. Super Admin mengklik button "Add New Admin"
4. Super Admin mengisi field admin tetapi tidak lengkap
5. Super Admin mengklik button "Create Admin"

Expected Result:

Sistem akan menampilkan pesan error "Please fill out this field!" pada field yang belum terisi.

Pengujian Kapasitas (SRS_NF-BALLS-002)

Pengujian Kapasitas (PV_BALLS_21)

Test Case: Sistem Menampung Semua Data Yang Masuk

Prosedur:

1. Penguji mengakses sistem BALLS
2. Penguji menambahkan data booking, user, payment, dan loyalty dalam jumlah besar
3. Penguji memantau performa database

Expected Result:

Sistem akan menampung semua data yang dimasukkan ke dalam database tanpa adanya batasan maksimal data dan tetap berfungsi dengan baik.

Pengujian Performa (SRS_NF-BALLS-003)

Pengujian Performa (PV_BALLS_22)

Test Case: Sistem Mampu Memberikan Layanan Secara Cepat Selama 2 Detik

Prosedur:

1. Penguji mengakses sistem BALLS
2. Penguji mencoba semua halaman dan proses yang tersedia pada sistem
3. Penguji mengukur waktu loading setiap halaman dan proses

Expected Result:

Sistem akan mengakses per halaman dan proses secara cepat maksimal selama 2 detik saja untuk memberikan user experience yang optimal.