# Test Cases + Edge Cases Document

## Promotion Engine for Order Pricing System

## 1. Objective

เอกสารนี้กำหนด Test Cases และ Edge Cases สำหรับระบบ **Promotion Engine** ที่ใช้คำนวณราคาสุทธิของคำสั่งซื้อ

โจทย์หลักที่ต้องทดสอบ:

```
Product 1 ลด 10%
Product 2 ลด 100 บาท
รองรับ Promotion หลายแบบ
รองรับ Promotion แบบซ้อนกัน
เพิ่ม Promotion ใหม่โดยไม่กระทบ Logic เดิม
คำนวณราคาสุทธิได้ถูกต้อง
```

เป้าหมายของ Test คือทำให้มั่นใจว่า:

- ระบบคำนวณส่วนลดถูกต้อง
- Promotion หลายตัวทำงานร่วมกันได้ถูกต้อง
- Promotion ซ้อนกันมีลำดับคำนวณชัดเจน
- Promotion Conflict ถูกจัดการได้
- ราคาสุทธิไม่ติดลบ
- Config ผิดไม่ทำให้ระบบพัง
- Confirm Order ปลอดภัยต่อ Concurrent Request
- Calculation Breakdown ตรวจสอบย้อนหลังได้
- เพิ่ม Strategy ใหม่แล้ว Logic เดิมไม่พัง

---

# 2. Test Strategy Overview

ควรแบ่งการทดสอบเป็น 6 ระดับ:

| Test Type | Purpose |
|---|---|
| Unit Test | ทดสอบ Logic ย่อย เช่น Strategy, Sorter, Stacking |
| Integration Test | ทดสอบหลาย Component ร่วมกัน |
| API Test | ทดสอบ HTTP Contract และ Error Response |
| Edge Case Test | ทดสอบกรณีผิดปกติ |
| Concurrency Test | ทดสอบ Usage Limit และ Race Condition |
| Regression Test | ป้องกัน Logic เดิมพังเมื่อเพิ่ม Promotion ใหม่ |

---

# 3. Test Data Setup

## 3.1 Product Test Data

| Product ID | Name | Price | `priceAmount` |
|---|---|---|---|
| 1 | Product 1 | 1,000 THB | 100000 |
| 2 | Product 2 | 500 THB | 50000 |
| 3 | Product 3 | 80 THB | 8000 |
| 4 | Product 4 | 2,000 THB | 200000 |

> ระบบเก็บเงินเป็น satang: 100 บาท = 10000

---

## 3.2 Base Promotion Data

### Promotion A: Product 1 ลด 10%

```
promotion_id = 1
code = ITEM1_10_PERCENT
scope = ITEM
target = PRODUCT 1
action_type = PERCENTAGE_DISCOUNT
value_basis_points = 1000
priority = 10
stackable = true
exclusive = false
stop_processing = false
status = ACTIVE
```

### Promotion B: Product 2 ลด 100 บาท

```
promotion_id = 2
code = ITEM2_MINUS_100
scope = ITEM
target = PRODUCT 2
action_type = FIXED_AMOUNT_DISCOUNT
value_amount = 10000
priority = 10
stackable = true
exclusive = false
stop_processing = false
status = ACTIVE
```

### Promotion C: Cart ซื้อครบ 1,000 ลด 100

```
promotion_id = 3
code = CART_1000_MINUS_100
scope = CART
target = CART
condition = MIN_ORDER_AMOUNT >= 100000
action_type = FIXED_AMOUNT_DISCOUNT
value_amount = 10000
priority = 20
stackable = true
status = ACTIVE
```

---

# 4. Unit Test Cases

## 4.1 PercentageDiscountStrategy

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-U001 | คำนวณลด 10% ได้ถูกต้อง | amount=100000, basisPoints=1000 | discount=10000, final=90000 |
| TC-U002 | ลด 0% | basisPoints=0 | skipped หรือ invalid_config |
| TC-U003 | ลดเกิน 100% | basisPoints=15000 | skipped reason=`invalid_percentage` |
| TC-U004 | มี Max Discount | amount=300000, 10%, max=20000 | discount=20000 |
| TC-U005 | Amount เป็น 0 | amount=0, 10% | discount=0, final=0 |
| TC-U006 | Amount ติดลบ | amount=-1000 | error หรือ invalid_context |

## 4.2 FixedAmountDiscountStrategy

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-U007 | ลด Fixed Amount ได้ถูกต้อง | amount=50000, discount=10000 | final=40000 |
| TC-U008 | Discount มากกว่ายอด | amount=8000, discount=10000 | discount=8000, final=0 |
| TC-U009 | Fixed Discount เป็น 0 | valueAmount=0 | skipped reason=`invalid_discount_amount` |
| TC-U010 | Fixed Discount ติดลบ | valueAmount=-10000 | skipped reason=`invalid_discount_amount` |

## 4.3 Discount Cap

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-U011 | Discount น้อยกว่า Amount | amount=100000, discount=10000 | actual=10000 |
| TC-U012 | Discount มากกว่า Amount | amount=8000, discount=10000 | actual=8000 |
| TC-U013 | Discount เป็น 0 | amount=100000, discount=0 | actual=0 |
| TC-U014 | Discount ติดลบ | amount=100000, discount=-100 | actual=0 หรือ invalid_config |

## 4.4 PromotionSorter

Sort Rule:

```
scope_order ASC
priority ASC
created_at ASC
promotion_id ASC
```

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-U015 | Scope ITEM มาก่อน CART | CART priority 1, ITEM priority 10 | ITEM ก่อน CART |
| TC-U016 | Priority น้อยกว่าทำก่อน | priority 1 และ priority 10 | priority 1 ก่อน |
| TC-U017 | Priority เท่ากันใช้ created_at | created_at เก่ากว่า | เก่ากว่าก่อน |
| TC-U018 | created_at เท่ากันใช้ ID | id 1 และ id 2 | id 1 ก่อน |
| TC-U019 | Sort ต้อง Deterministic | Input เดิมหลายครั้ง | Output order เหมือนเดิมทุกครั้ง |

## 4.5 StackingPolicy

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-U020 | Stackable true ใช้ร่วมกันได้ | Promo A/B stackable=true | Apply ได้ทั้งสอง |
| TC-U021 | Non-stackable Conflict | Promo A applied, Promo B stackable=false | Promo B skipped |
| TC-U022 | Exclusive Promotion | Promo A exclusive=true | Stop conflict scope |
| TC-U023 | Stop Processing | Promo A stop_processing=true | Promo ถัดไป skipped |
| TC-U024 | Scope ถูก Stop แล้ว | stoppedScope=ITEM | Promo ITEM skipped |
| TC-U025 | Item-level Stop เฉพาะ Item | Product 1 exclusive | ไม่ควรหยุด Product 2 |

# 5. Integration Test Cases

## 5.1 Basic Calculation

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-I001 | Product 1 ลด 10%, Product 2 ลด 100 | Product 1 qty=1, Product 2 qty=1 | final_total=130000 |

รายละเอียด Expected:

```
Product 1:
100000 - 10000 = 90000

Product 2:
50000 - 10000 = 40000

Original Total = 150000
Discount Total = 20000
Final Total = 130000
```

## 5.2 No Promotion Matched

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-I002 | Product ไม่มี Promotion | Product 4 qty=1 | original=200000, discount=0, final=200000 |

## 5.3 Quantity มากกว่า 1

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-I003 | Product 1 qty=3 ลด 10% | Product 1 qty=3 | original=300000, discount=30000, final=270000 |

## 5.4 Cart-level หลัง Item-level

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-I004 | Item Discount แล้ว Cart Discount | Product 1 + Product 2 + Cart Discount | final_total=120000 |

Calculation:

```
Product 1 final = 90000
Product 2 final = 40000
Cart subtotal = 130000
```

```
Cart discount = 10000
Final total = 120000
```

## 5.5 Cart Condition ไม่ผ่าน

| Test ID | Scenario | Input | Expected Result |
|---|---|---|---|
| TC-I005 | ยอดไม่ถึงขั้นต่ำ | Product 2 qty=1 | Cart Promo skipped reason= `min_order_amount_not_reached` |

# 6. API Test Cases

## 6.1 Pricing Calculate API

| Test ID | Endpoint | Scenario | Expected HTTP | Expected Result |
|---|---|---|---|---|
| TC-A001 | POST `/pricing/calculate` | Calculate สำเร็จ | 200 | final_total=130000 |
| TC-A002 | POST `/pricing/calculate` | ไม่มี items | 422 | code= `EMPTY_ORDER_ITEMS` |
| TC-A003 | POST `/pricing/calculate` | quantity=0 | 422 | code= `INVALID_QUANTITY` |
| TC-A004 | POST `/pricing/calculate` | product_id ไม่พบ | 404 | code= `PRODUCT_NOT_FOUND` |
| TC-A005 | POST `/pricing/calculate` | product inactive | 422 | code= `PRODUCT_INACTIVE` |
| TC-A006 | POST `/pricing/calculate` | currency ต่างกัน | 422 | code= `CURRENCY_MISMATCH` |
| TC-A007 | POST `/pricing/calculate` | invalid JSON | 400 | code= `INVALID_JSON` |

## 6.2 Promotion Admin API

| Test ID | Endpoint | Scenario | Expected HTTP | Expected Result |
|---|---|---|---|---|
| TC-A008 | POST `/promotions` | สร้าง Promotion ถูกต้อง | 201 | status= `DRAFT` |
| TC-A009 | POST `/promotions` | code ซ้ำ | 409 | code= `PROMOTION_CODE_ALREADY_EXISTS` |
| TC-A010 | POST `/promotions` | percentage > 100% | 422 | code= `INVALID_PROMOTION_CONFIG` |
| TC-A011 | POST `/promotions` | fixed amount ติดลบ | 422 | code= `INVALID_PROMOTION_CONFIG` |
| TC-A012 | POST `/promotions/{id}/activate` | Config ไม่ผ่าน | 422 | code= `PROMOTION_CONFIGURATION_INVALID` |
| TC-A013 | POST `/promotions/{id}/activate` | Version ไม่ตรง | 409 | code= `PROMOTION_VERSION_CONFLICT` |

## 6.3 Confirm Order API

| Test ID | Endpoint | Scenario | Expected HTTP | Expected Result |
|---|---|---|---|---|
| TC-A014 | POST `/orders/confirm` | Confirm สำเร็จ | 201 | order status= `CONFIRMED` |
| TC-A015 | POST `/orders/confirm` | ไม่มี Idempotency-Key | 400 | code= `IDEMPOTENCY_KEY_REQUIRED` |

| Test ID | Endpoint | Scenario | Expected HTTP | Expected Result |
|---|---|---|---|---|
| TC-A016 | POST `/orders/confirm` | ราคาเปลี่ยนจาก preview | 409 | code= `ORDER_PRICE_CHANGED` |
| TC-A017 | POST `/orders/confirm` | usage limit เต็ม | 409 | code= `PROMOTION_USAGE_LIMIT_REACHED` |

# 7. Stacking Test Cases

## 7.1 Stackable Promotions

| Test ID | Scenario | Promotions | Expected |
|---|---|---|---|
| TC-S001 | ลด 10% แล้วลด 100 | A priority=1, B priority=2 | 100000 → 90000 → 80000 |

## 7.2 Percentage ซ้อน Percentage

| Test ID | Scenario | Promotions | Expected |
|---|---|---|---|
| TC-S002 | ลด 10% แล้วลด 10% | A priority=1, B priority=2 | final=81000 |

Calculation:

```
100000 - 10% = 90000
90000 - 10% = 81000
```

ต้องไม่คำนวณจาก Original ทั้งสองรอบ

## 7.3 Priority กลับกันแล้วผลลัพธ์เปลี่ยน

| Test ID | Scenario | Promotions | Expected |
|---|---|---|---|
| TC-S003 | Fixed ก่อน Percentage | fixed 10000 priority=1, percent 10% priority=2 | final=81000 |
| TC-S004 | Percentage ก่อน Fixed | percent 10% priority=1, fixed 10000 priority=2 | final=80000 |

จุดประสงค์:

```
พิสูจน์ว่า Priority มีผล และระบบต้อง Deterministic
```

## 7.4 Exclusive Promotion

| Test ID | Scenario | Expected |
|---|---|---|
| TC-S005 | Promo A exclusive=true, Promo B same conflict group | Promo A applied, Promo B skipped |
| TC-S006 | Exclusive เฉพาะ Product 1 | Product 2 ยัง apply promotion ได้ |

## 7.5 Stop Processing

| Test ID | Scenario | Expected |
|---|---|---|
| TC-S007 | Promo A stop_processing=true | Promo B skipped reason= `scope_stopped` |

### 7.6 Non-stackable

| Test ID | Scenario | Expected |
|---|---|---|
| TC-S008 | Promo B stackable=false หลัง Promo A applied | Promo B skipped reason= `non_stackable_conflict` |

## 8. Condition / Target Edge Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-E001 | Product Target ตรง | Promotion applied |
| TC-E002 | Product Target ไม่ตรง | skipped reason= `target_not_matched` |
| TC-E003 | Category Target ตรง | Promotion applied |
| TC-E004 | Category Target ไม่ตรง | skipped reason= `target_not_matched` |
| TC-E005 | Coupon Code ตรง | Promotion applied |
| TC-E006 | Coupon Code ไม่ตรง | skipped reason= `coupon_not_matched` |
| TC-E007 | Promotion ต้องใช้ Coupon แต่ไม่มี Coupon | skipped reason= `coupon_required` |
| TC-E008 | Min Order Amount ถึง | Promotion applied |
| TC-E009 | Min Order Amount ไม่ถึง | skipped reason= `min_order_amount_not_reached` |
| TC-E010 | Condition AND ผ่านทั้งหมด | Promotion applied |
| TC-E011 | Condition AND ไม่ผ่านบางตัว | Promotion skipped |
| TC-E012 | Condition OR ผ่านอย่างน้อยหนึ่ง | Promotion applied |

## 9. Date / Status Edge Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-D001 | status=ACTIVE และอยู่ในช่วงเวลา | eligible |
| TC-D002 | status=INACTIVE | skipped reason= `inactive` |
| TC-D003 | starts_at > now | skipped reason= `not_started` |
| TC-D004 | ends_at < now | skipped reason= `expired` |
| TC-D005 | starts_at == now | eligible |
| TC-D006 | ends_at == now | eligible |
| TC-D007 | timezone ต่างกัน | แปลงเป็นเวลามาตรฐานก่อนเทียบ |

Rule ที่ต้องใช้:

```
starts_at <= now AND ends_at >= now
```

## 10. Invalid Config Edge Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-C001 | action_type ไม่มี Strategy | skipped reason= `strategy_not_found` , system ไม่ panic |
| TC-C002 | FIXED_AMOUNT ไม่มี value_amount | skipped reason= `invalid_promotion_config` |
| TC-C003 | PERCENTAGE ไม่มี value_basis_points | skipped reason= `invalid_promotion_config` |
| TC-C004 | value_basis_points > 10000 | skipped reason= `invalid_percentage` |
| TC-C005 | value_amount < 0 | skipped reason= `invalid_discount_amount` |
| TC-C006 | Promotion ไม่มี Action | skipped reason= `action_not_found` |
| TC-C007 | ITEM scope ไม่มี Target | skipped reason= `target_required` |
| TC-C008 | value_json parse ไม่ได้ | rejected ตอนสร้าง Promotion หรือ skipped runtime |
| TC-C009 | Condition Type ไม่รองรับ | skipped reason= `condition_not_supported` |

# 11. Money / Rounding Edge Cases

| Test ID | Scenario | Input | Expected |
|---|---|---|---|
| TC-M001 | ใช้ satang แล้วคำนวณถูก | amount=100000, 10% | discount=10000 |
| TC-M002 | Percentage ได้เศษ | amount=9999, 10% | discount=999 ถ้าใช้ floor |
| TC-M003 | Final total ห้ามติดลบ | original=8000, discount=10000 | final=0 |
| TC-M004 | Currency ต่างกัน | THB + USD | calculation failed |
| TC-M005 | Product ราคา 0 | price=0 | final=0, discount=0 |
| TC-M006 | Quantity ใหญ่มาก | qty มากผิดปกติ | reject หรือ overflow protection |
| TC-M007 | Integer overflow | amount * basisPoints เกิน limit | error= `CALCULATION_OVERFLOW` |

# 12. Calculation Breakdown Test Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-B001 | Applied Promotion ต้องมี before/discount/after | มีครบทุก field |
| TC-B002 | Skipped Promotion ต้องมี reason | มี reason ที่ชัดเจน |
| TC-B003 | Item-level breakdown ถูกต้อง | item.discount = sum item discounts |
| TC-B004 | Order-level breakdown ถูกต้อง | order.discount_total ถูกต้อง |
| TC-B005 | Stacking ต้องแสดงลำดับ | Applied promotions เรียงตาม execution order |
| TC-B006 | Calculation policy ต้องแสดง | มี scopeOrder, priorityOrder, roundingMode |

# 13. Confirm Order / Consistency Test Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-O001 | Calculate แล้ว Promotion ถูกปิดก่อน Confirm | Confirm ต้อง Recalculate และไม่ใช้ Promotion นั้น |
| TC-O002 | Product Price เปลี่ยนก่อน Confirm | Return `ORDER_PRICE_CHANGED` |
| TC-O003 | Confirm ต้องสร้าง Order และ Order Items | Rows ถูกสร้างครบ |
| TC-O004 | Order Item ต้อง Snapshot ราคาเดิม | ราคาเก่าไม่เปลี่ยนตาม Product ปัจจุบัน |
| TC-O005 | Confirm ต้อง Consume Usage | promotion_usages เพิ่ม |

| Test ID | Scenario | Expected |
|---|---|---|
| TC-O006 | Confirm ซ้ำด้วย Idempotency-Key เดิม | คืนผลเดิม ไม่สร้าง Order ซ้ำ |
| TC-O007 | Idempotency-Key เดิมแต่ Payload ต่าง | Return `IDEMPOTENCY_KEY_CONFLICT` |

# 14. Concurrency Test Cases

| Test ID | Scenario | Action | Expected |
|---|---|---|---|
| TC-CC001 | Coupon เหลือ 1 สิทธิ์ ยิง Confirm 10 request พร้อมกัน | Concurrent confirm | สำเร็จ 1, ล้มเหลว 9 |
| TC-CC002 | User ใช้ Coupon ได้ครั้งเดียว | Confirm 2 ครั้ง user เดิม | ครั้งที่ 2 fail หรือ skipped |
| TC-CC003 | Calculate พร้อมกันไม่ Consume Usage | 100 concurrent calculate | usage ไม่เพิ่ม |
| TC-CC004 | Promotion Usage Lock ทำงาน | concurrent transaction | ไม่มี usage เกิน limit |
| TC-CC005 | DB Transaction Rollback | error ระหว่าง confirm | ไม่สร้าง partial order |

# 15. Performance / Scale Test Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-P001 | 100 Items, 100 Promotions | จบภายใน Threshold |
| TC-P002 | 2 Items, 1000 Active Promotions | Filter เฉพาะที่เกี่ยวข้อง |
| TC-P003 | Query Count Test | ไม่มี Query ใน Loop |
| TC-P004 | Cache Hit Active Promotions | ลด Query MySQL |
| TC-P005 | Cache Down | fallback DB ได้ |
| TC-P006 | Calculation Timeout | return `CALCULATION_TIMEOUT` |
| TC-P007 | Large Breakdown | Response ยังไม่ใหญ่เกิน Limit |

Expected Query Pattern:

```
1 query load products by IDs
1 query group load active promotions with relations
1 query save calculation log
```

ต้องไม่เป็น:

```
N queries per item
N queries per promotion
```

# 16. Security / Authorization Test Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-SEC001 | Customer เรียก Admin API | 403 FORBIDDEN |
| TC-SEC002 | ไม่มี Token | 401 UNAUTHORIZED |
| TC-SEC003 | Customer ดู Order คนอื่น | 403 ORDER_ACCESS_DENIED |
| TC-SEC004 | Client ส่ง price มาใน request | Server ignore หรือ reject |

| Test ID | Scenario | Expected |
|---|---|---|
| TC-SEC005 | Explain API ถูกเรียกโดย Customer | 403 FORBIDDEN |
| TC-SEC006 | SQL Injection ใน query | ไม่กระทบ DB |
| TC-SEC007 | Request body ใหญ่มาก | 413 PAYLOAD_TOO_LARGE |

---

# 17. API Error Response Test Cases

| Test ID | Scenario | Expected |
|---|---|---|
| TC-R001 | Invalid JSON | 400 `INVALID_JSON` |
| TC-R002 | Missing required field | 400 `MISSING_REQUIRED_FIELD` |
| TC-R003 | Duplicated product_id | Merge quantity หรือ reject ตาม policy |
| TC-R004 | Error response ต้องมี request_id | มี meta.requestId |
| TC-R005 | Unknown server error | 500 ไม่คืน stack trace |
| TC-R006 | Rate limit | 429 `TOO_MANY_REQUESTS` |

Recommendation สำหรับ duplicated product:

```
MVP แนะนำ Merge Quantity เพื่อ User-friendly
แต่ต้องกำหนดชัดใน API Contract
```

---

# 18. Regression Test Cases

ทุกครั้งที่เพิ่ม Promotion Strategy ใหม่ เช่น `BUY_X_GET_Y` ต้อง Run Regression Set:

| Test ID | Scenario |
|---|---|
| TC-REG001 | Product 1 ลด 10% ยังถูกต้อง |
| TC-REG002 | Product 2 ลด 100 บาทยังถูกต้อง |
| TC-REG003 | Item-level ก่อน Cart-level ยังถูกต้อง |
| TC-REG004 | Stacking ยังถูกต้อง |
| TC-REG005 | Exclusive ยังหยุดถูก Scope |
| TC-REG006 | Stop Processing ยังทำงาน |
| TC-REG007 | Discount Cap ยังป้องกันราคาติดลบ |
| TC-REG008 | Breakdown ยังครบ |
| TC-REG009 | Strategy Not Found ไม่ Panic |
| TC-REG010 | Existing API Contract ไม่เปลี่ยน |

---

# 19. Minimum Test Set สำหรับ MVP

ถ้าเวลาจำกัด ควรทำชุดนี้ก่อน:

| Priority | Test ID | Scenario |
|---|---|---|
| P0 | TC-U001 | Percentage Discount |

| Priority | Test ID | Scenario |
|---|---|---|
| P0 | TC-U007 | Fixed Amount Discount |
| P0 | TC-U008 | Discount มากกว่าราคา |
| P0 | TC-U015 | Sort by Scope |
| P0 | TC-U016 | Sort by Priority |
| P0 | TC-U022 | Exclusive Promotion |
| P0 | TC-I001 | Basic Calculation |
| P0 | TC-I003 | Quantity > 1 |
| P0 | TC-I004 | Cart Discount after Item |
| P0 | TC-A001 | Calculate API Success |
| P0 | TC-A003 | Invalid Quantity |
| P0 | TC-S001 | Stackable Promotion |
| P0 | TC-D004 | Expired Promotion |
| P0 | TC-C001 | Strategy Not Found |
| P0 | TC-B001 | Applied Breakdown |
| P0 | TC-O001 | Confirm Recalculate |
| P0 | TC-CC003 | Calculate Does Not Consume Usage |

# 20. Final Demo Test Case

## Scenario

```
Product 1 ราคา 1,000 บาท
Product 2 ราคา 500 บาท

Promotion 1:
Product 1 ลด 10%

Promotion 2:
Product 2 ลด 100 บาท
```

## Request

```
{
  "items": [
    {
      "productId": 1,
      "quantity": 1
    },
    {
      "productId": 2,
      "quantity": 1
    }
```

```
  ]
}
```

## Expected Calculation

```
Product 1:
1,000 - 10% = 900

Product 2:
500 - 100 = 400

Final:
900 + 400 = 1,300
```

## Expected Response

```
{
  "originalTotal": 150000,
  "discountTotal": 20000,
  "finalTotal": 130000
}
```

## Expected Breakdown

```
applied_promotions:
- ITEM1_10_PERCENT discount 10000
- ITEM2_MINUS_100 discount 10000
```

---

# 21. Acceptance Criteria

ระบบถือว่าผ่าน Test เมื่อ:

- Basic Calculation ถูกต้อง
- Percentage Discount ถูกต้อง
- Fixed Amount Discount ถูกต้อง
- Quantity มากกว่า 1 ถูกต้อง
- Discount ไม่ทำให้ราคาติดลบ
- Promotion expired/inactive ถูก Skip
- Promotion Priority ทำงานถูก
- Promotion Stacking ทำงานถูก
- Exclusive และ Stop Processing ทำงานถูก
- Target และ Condition ทำงานถูก
- Calculation Breakdown ครบ

- Calculation Log ถูกบันทึก
- Confirm Order Recalculate ก่อนสร้าง Order
- Calculate API ไม่ Consume Usage
- Concurrency Usage Limit ไม่เกิน Quota
- Invalid Config ไม่ทำให้ระบบ Panic
- ไม่มี DB Query ใน Loop item/promotion
- เพิ่ม Promotion ใหม่แล้ว Regression Test ผ่าน

---

## Final Test Summary

Test Strategy ของ Promotion Engine ต้องไม่ทดสอบแค่ว่า “ลดราคาได้ไหม” แต่ต้องทดสอบว่า:

```
ลดถูกไหม
ลดตามลำดับไหม
ซ้อนกันถูกไหม
หยุด Rule ถูกไหม
Config ผิดแล้วระบบไม่พังไหม
Usage Limit ถูกไหม
Concurrent Request ปลอดภัยไหม
Response Debug ได้ไหม
Order จริง Recalculate ไหม
เพิ่ม Promotion ใหม่แล้วของเดิมไม่พังไหม
```

ถ้าครอบคลุม Test เหล่านี้ ระบบจะมีความน่าเชื่อถือระดับ Production มากกว่าการทำแค่ Prototype.