

Nama : Rafie aydin ihsan

Nim: 1302220065

Kelas : se-46-04

Requirement of subprogram to be tested:

Input specification: the subprogram takes an integer input in the range $[-100, 100]$

Functional specification: Output is the sign of the input value (value 0 is considered positive)

1.

- **Spesifikasi Input:** Bilangan bulat dalam rentang $[-100, 100]$.
- **Spesifikasi Fungsional:** Keluaran adalah tanda input, dengan 0 dianggap positif.

Berdasarkan spesifikasi, keluaran tergantung pada apakah input kurang dari 0 atau tidak, dengan 0 secara eksplisit dianggap positif. Ini menciptakan dua partisi logis:

1. **Partisi 1:** $\text{Input} < 0$ (bilangan negatif, rentang -100 hingga -1).
2. **Partisi 2:** $\text{Input} \geq 0$ (nol dan bilangan positif, rentang 0 hingga 100).

Format keluaran tidak disebutkan secara eksplisit, menimbulkan ambiguitas. Kemungkinan interpretasi meliputi:

- Keluaran berupa string: "positive" untuk $\text{input} \geq 0$, "negative" untuk $\text{input} < 0$.
- Keluaran berupa bilangan bulat: 1 untuk $\text{input} \geq 0$, -1 untuk $\text{input} < 0$, yang sesuai dengan fungsi tanda standar yang dimodifikasi untuk masalah ini.

Dalam konteks pemrograman, keluaran numerik lebih umum untuk fungsi tanda, terutama karena spesifikasi menyebutkan "tanda input" dan 0 dianggap positif, yang sesuai dengan keluaran 1 untuk $\text{input} \geq 0$ dan -1 untuk $\text{input} < 0$. Namun, dalam analisis awal, ada pertimbangan bahwa keluaran mungkin berupa string berdasarkan interpretasi teks, tetapi akhirnya diputuskan untuk menggunakan keluaran numerik berdasarkan praktik pemrograman standar.

Menurut EQP, pilih satu kasus uji per partisi:

- Untuk Partisi 1 ($\text{input} < 0$), pilih bilangan negatif, misalnya -1, dengan keluaran diharapkan -1.
- Untuk Partisi 2 ($\text{input} \geq 0$), pilih 0 untuk memverifikasi kasus khusus, dengan keluaran diharapkan 1, memastikan 0 ditangani sebagai positif.

Alternatifnya, bisa memilih bilangan positif seperti 1 untuk Partisi 2, tetapi menguji 0 secara eksplisit lebih sesuai dengan penekanan masalah pada 0.

Selain itu, analisis nilai batas juga dipertimbangkan:

- Batas bawah: -100 (input < 0 , keluaran diharapkan -1).
- Batas atas: 100 (input ≥ 0 , keluaran diharapkan 1).
- Titik kritis: 0 (input = 0, keluaran diharapkan 1, mengonfirmasi klasifikasi positif).

Berdasarkan EQP dan kebutuhan respons cepat, kasus uji adalah:

- **Kasus Uji 1:** Input -1, Keluaran Diharapkan -1 (mencakup partisi input < 0 , dekat batas).
- **Kasus Uji 2:** Input 0, Keluaran Diharapkan 1 (mencakup partisi input ≥ 0 , memverifikasi kasus khusus 0).

Requirement of subprogram to be tested:

Input specification: the subprogram takes an integer input in the range $[-100, 100]$

Functional specification: Output is the sign of the input value (value 0 is considered positive)

Test Case using BVA?

2.

Berdasarkan spesifikasi fungsional:

- Partisi 1: Input < 0 (output "negatif"). Dalam rentang $[-100, 100]$, ini adalah dari -100 hingga -1.
- Partisi 2: Input ≥ 0 (output "positif"). Dalam rentang $[-100, 100]$, ini adalah dari 0 hingga 100.

Batas kritis adalah:

- Batas bawah rentang input: -100.
- Batas atas rentang input: 100.
- Batas partisi fungsional: 0, di mana output berubah dari "negatif" menjadi "positif".

Untuk memastikan pengujian menyeluruh, perlu menguji:

- Nilai di batas rentang input (-100 dan 100).

- Nilai di sekitar batas partisi fungsional (0), yaitu -1 (di bawah 0) dan 1 (di atas 0) untuk memverifikasi transisi output.

Pemilihan Kasus Uji

Berdasarkan prinsip BVA dan analisis di atas, kasus uji dipilih sebagai berikut:

- **-100:** Menguji batas bawah rentang input dan partisi input < 0 . Output diharapkan "negatif".
- **-1:** Menguji nilai terbesar dalam partisi input < 0 , tepat di bawah 0, untuk memastikan output tetap "negatif".
- **0:** Menguji batas partisi fungsional, di mana input ≥ 0 , output diharapkan "positif" (sesuai spesifikasi bahwa 0 dianggap positif).
- **1:** Menguji nilai di atas batas partisi fungsional, untuk memastikan output "positif" dan subprogram menangani angka positif dengan benar.
- **100:** Menguji batas atas rentang input dan partisi input ≥ 0 , output diharapkan "positif".

Pilihan ini mencakup:

- Batas rentang input (-100 dan 100).
- Batas partisi fungsional (0) dan nilai di sekitarnya (-1 dan 1) untuk memastikan transisi output benar.

The software module for calculates entrance ticket prices for the Golden Karaoke. The Center's ticket price depends on 3 variables: day (weekday, weekend), visitor's status (M = member, NM = non-member), entry hour (11.00–18.00, 18,01-22.00). Use Decision Table Testing for ticket prices module and make test cases tables that correspond to that DT. You can use any assumptions which is needed (tickect prices, discount for member, happy hour etc).

3.

Harga dasar untuk non-member:

- Weekday, 11:00-18:00: \$20,00 (dianggap off-peak dengan harga lebih rendah).
 - Weekday, 18:01-22:00: \$25,00 (surcharge untuk jam malam).
 - Weekend, 11:00-18:00: \$25,00 (harga lebih tinggi karena permintaan lebih besar).
 - Weekend, 18:01-22:00: \$30,00 (kombinasi weekend dan jam malam).
- Member mendapatkan diskon 20% dari harga non-member untuk kategori yang sama, yang dihitung sebagai 80% dari harga non-member. Misalnya, untuk weekday 11:00-18:00, harga member adalah $\$20,00 * 0,8 = \$16,00$.

Berikut adalah tabel yang mencakup semua kasus uji:

No	Hari	Status	Jam Masuk	Harga ticket
1	Weekday	Member	11:00 - 18:00	\$16,00
2	Weekday	Member	18:01 - 22:00	\$20,00
3	Weekday	Non-Member	11:00 - 18:00	\$20,00
4	Weekday	Non-Member	18:01 - 22:00	\$25,00
5	Weekday	Member	11:00 - 18:00	\$20,00
6	Weekend	Member	18:01 - 22:00	\$24,00
7	Weekend	Non-member	11:00 - 18:00	\$25,00
8	Weekend	Non-member	18:01 - 22:00	\$30,00

Setiap baris diverifikasi untuk memastikan perhitungan harga member benar, misalnya:

- Untuk aturan 1, harga non-member weekday 11:00-18:00 adalah \$20,00, jadi member $\$20,00 * 0,8 = \$16,00$.
- Untuk aturan 6, harga non-member weekend 18:01-22:00 adalah \$30,00, jadi member $\$30,00 * 0,8 = \$24,00$.

Semua perhitungan konsisten dengan asumsi awal.

Setiap baris dalam tabel keputusan juga berfungsi sebagai kasus uji. Misalnya:

- Kasus Uji 1: Input: Hari = Weekday, Status = Member, Jam Masuk = 11:00-18:00; Output Diperkirakan: \$16,00.
- Kasus Uji 8: Input: Hari = Weekend, Status = Non-member, Jam Masuk = 18:01-22:00; Output Diperkirakan: \$30,00.