**Uji White Box**

**Teknik Decision Table**

Pada gambar ada dua garis yang membagi tabel menjadi struktur utamanya. Garis vertikal solid memisahkan bagian Stub dan Entry dari tabel, dan garis horizontal solid adalah batas antara Conditions dan Actions.

Jadi baris-baris ini memisahkan tabel menjadi empat bagian, Condition Stub, Action Stub, Condition Entries, dan Action Entries.

Kolom di bagian entri tabel dikenal sebagai aturan.

Nilai yang berada di kolom entri kondisi dikenal sebagai input.

Nilai di dalam bagian entri tindakan dikenal sebagai output.

Output dihitung tergantung pada input dan spesifikasi program.

Input dalam tabel yang diberikan ini menghasilkan output tergantung pada kondisi apa yang dipenuhi input ini. Perhatikan penggunaan "-" pada tabel di bawah, ini dikenal sebagai entri tidak peduli. Entri Don't care biasanya dianggap sebagai nilai palsu yang tidak memerlukan nilai untuk menentukan output.

Gambar menunjukkan nilainya dari input sebagai nilai true(T) atau false(F) yang merupakan kondisi biner, tabel yang menggunakan kondisi biner dikenal sebagai tabel keputusan entri terbatas. Tabel yang menggunakan beberapa kondisi dikenal sebagai tabel keputusan entri yang diperluas. Salah satu aspek penting yang perlu diperhatikan tentang tabel keputusan adalah tabel tersebut tidak penting karena tidak menerapkan urutan tertentu atas kondisi atau tindakan.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stub | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 |
| Path 3 (login) | T | T | T | F | - | T |
| Path 2 (absen) | F | T | T | T | - | T |
| A1 | X | X |  | X | X | X |
| A2 |  | X |  |  |  |  |
| A3 | X |  |  | X |  |  |
| A4 |  |  | X |  |  | X |

**Teknik BVA**

Partisi kesetaraan (EP) dan analisis nilai batas (BVA) memberikan strategi untuk menulis kasus pengujian white-box. Tidak diragukan lagi, setiap kali Anda menghadapi segala jenis nomor atau membatasi dalam persyaratan

Sebagai pengujian 1, seseorang ingin login,dengan memiliki email dan password. Mengingat EP / BVA, saya ingin memastikan kasus uji ini meliputi:

1. Memasukkan nilai email: admin@gmail.com dan Password: admin, Email dan Passwaord Benar (kelas kesetaraan “adalah seorang admin”)
2. Memasukkan nilai email: wahyu@gmail.com dan Password: wahyu, Email dan Passwaord Benar (kelas kesetaraan “adalah seorang pegawai”)
3. Memasukkan nilai email: admin@gmail.com dan Password: admin, Role User adalah 1 (kelas kesetaraan “adalah seorang admin”)
4. Memasukkan nilai email: wahyu@gmail.com dan Password: wahyu, Role User adalah selain 1 (kelas kesetaraan “adalah seorang pegawai”)
5. Memasukkan nilai email dan password salah (kelas kesetaraan “bukan user yang terdaftar”)

Sebagai pengujian 2, seseorang ingin melakukan absensi, serta admin konfimasi absensi tersebut. Mengingat EP / BVA, saya ingin memastikan kasus uji ini meliputi:

1. Memilih tombol ‘masuk’ (kelas kesetaraan “status kehadiran adalah masuk”)
2. Memilih tombol ‘izin’ (kelas kesetaraan “status kehadiran adalah izin”)
3. Memilih tombol ‘sakit’ (kelas kesetaraan “status kehadiran adalah sakit”)
4. Memilih tombol ‘konfirmasi’ (kelas kesetaraan “status kehadiran adalah masuk, izin, sakit”)
5. Memilih tombol ‘tolak’ (kelas kesetaraan “status kehadiran adalah tidak hadir”)
6. Memilih tombol ‘masuk’, ‘izin’, ‘sakit’ (kelas kesetaraan “status kehadiran adalah tidak hadir”)

Dengan loop pemrograman (seperti perulangan for), pertimbangkan EP dan melaksanakan loop di tengah operasional terikat. Untuk BVA, akan ingin memastikan bahwa menjalankan loop tepat di bawah, sudah tepat, dan tepat di atas kondisi batas.