**TUGAS**

**Rekayasa Perangkat Lunak**

**“APLIKASI PEMESANAN TIKET TRAVEL ”**

**METODE WHITEBOX & BLACKBOX**



**Nama : Dwi Hardiani C**

**NIM : 16090071**

**Kelas : 6C**

**Dosen Pengampu : La Ode Mohammad Zulfiqar,ST.,M.Kom**

**PROGRAM STUDI DIV TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**T E G A L**

**2019**

1. **WHITE BOX TESTING**
2. **Pengertian White Box Testing**

White box testing adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar. Pengujian dilakukan berdasarkan bagaimana suatu software menghasilkan output dari input. Pengujian ini dilakukan berdasarkan kode program. Disebut juga struktural testing atau glass box testing.

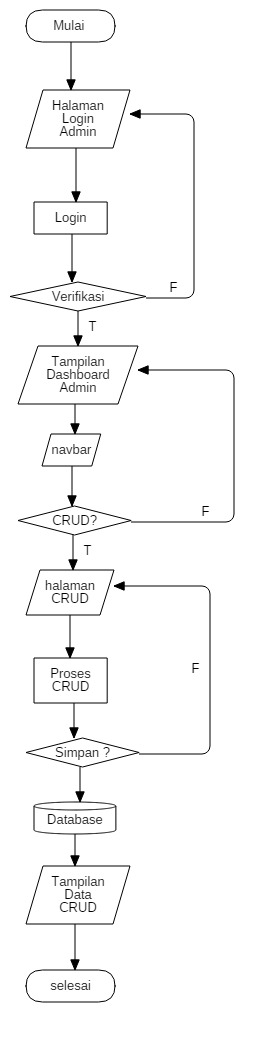
Teknik pengujian :

1. Notasi Diagram Alir (Flowchart)

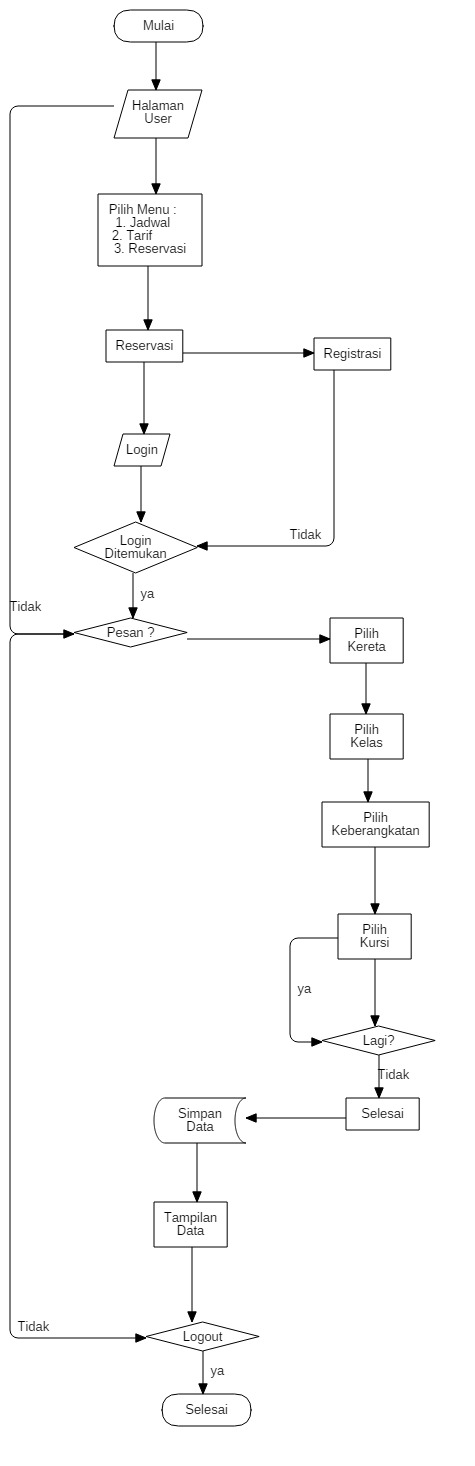
Adalah sebuah jenis diagram yang mewakili algoritma, alur kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol grafis dan urutannya dihubungkan dengan panah. Diagram ini mewakili ilustrasi atau penggambaran penyelesaian masalah. Diagram alir digunakan untuk menganalisis, mendesain, mendokumentasi atau memanajemen sebuah proses atau program di berbagai bidang.

Berikut ini diagram alir yang telah dibuat :

1. Flowchart Admin
2. Pertama kali admin harus login terlebih dahulu ke dalam admin pemesanan ini. Login menggunakan email dan memasukkan pasword masing-masing admin.
3. Lalu setelah login berhasil akan dibawa menuju ke halaman dashboard awal sistem, apabila login gagal maka akan kembali lagi ke halaman login.
4. Ada modul dashboard untuk kembali ke tampilan awal website.
5. Setelah itu admin bisa logout sistem dan kembali pada tampilan login kembali



1. Flowchart User
2. Pertama kali user harus masuk ke link website yang telah tersedia.
3. Setelah user berhasil masuk maka akan menampilkan dashboard utama yang berisi : jadwal, tarif, reservasi.
4. Setelah melihat dashboard utama maka akan menampilkan Reservasi dan bagian Registrasi .
5. Setelah itu masuk ke menu login. Login menggunakan nomor pemesanan yang tersedia.
6. Setelah login berhasil akan dibawa menuju ke halaman dashboard sistem untuk memilih kereta dan tempat duduk yang di inginkan , apabila loading gagal maka akan kembali ke menu login
7. Data yang diberikan akan dimasukkan ke dalam databasenya.
8. Setelah data berhasil tersimpan maka akan menampilkan pada Dashboard.



1. **Basic path (Control Flow Graph)**

Uji coba basis path adalah teknik uji coba white box yg diusulkan Tom McCabe. Metode ini memungkinkan perancang test case mendapatkan ukuran kekompleksan logical dari perancangan prosedural dan menggunkan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Test case yg didapat digunakan untuk mengerjakan basis set yg menjamin pengerjaan setiap perintah minimal satu kali selama uji coba.

Flow Graph merupakan grafik yang digunakan untuk menggambarkan aliran kontrol dari sebuah program. Berbeda dengan flowchart, grafik pada flow graph tidak menggambarkan secara detail proses yang terjadi pada setiap blok notasi. Jenis notasi pada flowchart digambarkan secara berbeda (diamond, persegi panjang, jajar genjang, dst) untuk menggambarkan proses yang berbeda, sedangkan notasi pada flow graph hanya diwakili oleh sebuah notasi lingkaran. Dari penggunaannya, flowchart digunakan pada tahapan perancangan untuk menggambarkan logika dari program sedangkan flow graph digunakan pada tahapan pengujian yang berfokus pada penggambaran aliran kontrol sebuah program.

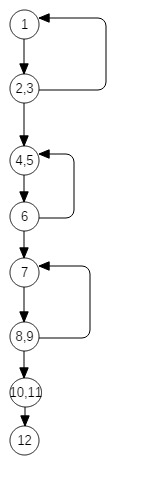
Notasi lingkaran disebut sebagai *flow graph****node*** yang digunakan untuk menggambarkan *statement-statement*berikut:

1. Satu atau lebih statement secara sekuensial yang dikelompokkan
2. Percabangan seleksi dari satu statement kedua pilihan statement (seleksi)
3. Penggabungan dua statement yang dilanjutkan pada satu statement yang sama (merge).

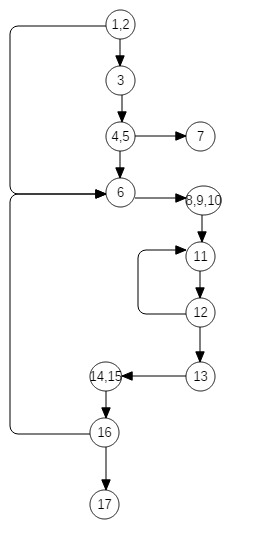
Sedangkan notasi garis panah disebut sebagai ***edge***atau ***link***, menggambarkan aliran kontrol. Setiap *edge*harus dihubungkan dari/kesebuah *node*, meskipun *node*tersebut tidak mewakili sebuah *statement*khusus. Area yang dibatasi oleh *node* dan *edge*disebut sebagai ***region***. Secara sederhana, flow graph dapat dibuat dari grafik flowchart ataupun dari pseudocode/program design language/source code yang telah dibuat sebelumnya.

Berikut ini Control Flow Graph yang telah terbentuk :

1. CFG untuk Admin



1. CFG untuk User



1. **Cyclomatic Complexity**

Cyclomatic Complexity merupakan suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. Pada Basis Path Testing, hasil dari cyclomatic complexity digunakan untuk menentukan banyaknya independent paths. Independent path adalah sebuah kondisi pada program yang menghubungkan node awal dengan node akhir.

Terdapat 2 persamaan yang digunakan, yaitu:

V(G)= E - N + 2 atau V(G)= P + 1

Keterangan:

V(G)= cyclomatic complexity untuk flow graph G

E=Jumlah edge(panah)

N=Jumlah node(lingkaran)

P=Jumlah predicate node

Rumus yang digunakan untuk menentukan Cyclocmatic Complexity adalah : **P + 1**

Maka dari itu Cyclocmatic Complexity yang terbentuk adalah sebagai berikut :

1. Bagian Admin...................................................... pemisah Seksi(Berkelanjut)

* (E – N) + 2 - Decision + 1

= (10-8) +2 = 3 + 1

= 2+2 = 4

= 4

1. Bagian User

* ( E – N) + 2 - Decision + 1

= (15 – 12) + 2 = 4 + 1

= 3 + 2 = 5

= 5

1. **Basic Path**
2. Bagian Admin

Path 1 : 1 – 2 – 3 - 1

Path 2 : 1 – 2 – 3 – 4 – 5 – 6 – 5 – 4

Path 3 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 7

Path 4 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 (v)

1. Bagian User

Path 1 : 1 – 2 – 3 – 4 – 5 – 6 – 2

Path 2 : 4 – 5 – 7

Path 3 : 6 – 8 – 9 – 10

Path 4 : 8 – 11 – 12 – 11

Path 5 : 6 – 14 – 15 – 6

Path 6 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17

* **Kelebihan White Box Testing**
* **Kesalahan logika.** Digunakan pada sintaks ‘if’ dan pengulangan. Dimana White Box Testing akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.
* **Ketidaksesuaian asumsi.** Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
* **Kesalahan ketik.**Mendeteksi bahasa pemrograman yang bersifat case sensitive.
* **Kelemahan White Box Testing**

Untuk perangkat lunak yang tergolong besar, White Box Testing dianggap sebagai strategi yang tergolong boros, karena akan melibatkan sumber daya yang besar untuk melakukannya

1. ***BLACK BOX TESTING***
2. **Pengertian *Black Box Testing***

*Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu koatak hitam, kit hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya(interface nya) , fungsionalitasnya.tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui input dan output).

*Black Box* pengujian adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian white-box). Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga.

Pengujian pada Black Box berusaha menemukan kesalahan seperti:

* Fungsi-fungsi yang tidak benar atau hilang
* Kesalahan interface
* Kesalahan dalam struktur data atau akses database eksternal
* Kesalahan kinerja
* Inisialisasi dan kesalahan terminasi
  1. **Teknik khas Black Box Testing desain meliputi:**

1. **Equivalent:**

Artikel : 100 digit

|  |  |  |  |
| --- | --- | --- | --- |
| invalid | valid | | Invalid |
| -100 - 0 | 1-100 | A-Z | 101-1000 |
| 0 | 100 | Z | 101 |

Komentar : 100 digit

|  |  |  |  |
| --- | --- | --- | --- |
| invalid | valid | | Invalid |
| -100 - 0 | 1-100 | A-Z | 101-1000 |
| 0 | 100 | Z | 101 |

1. **Tabel Kebenaran**
2. Validasi Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Username | F | F | T | T |
| Password | F | T | F | T |
| Hasil | eror | eror | Eror | benar |

1. Validasi :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Artikel | F | F | F | F | T |
| Komentar | F | F | F | T | T |
| Hasil | Eror | Eror | Eror | Eror | Benar |