|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Judul | pseudocode | harapan | hasil |
| 1 | Method menyimpan data ketika tombol simpan di tekan |  | Dari kode tersebut harapannya aplikasi dapat menyimpan data kedalam database | Valid |
| 2 | Method untuk memeriksa jadwal sholat |  | Dari kode tersebut hasil yang diharapkan adalah aplikasi dapat memeriksa pada jam sekian lalu sholat yang dilaksanakan sholat pada jam tersebut. Contoh pada jam 12.30 wib yang tampil adalah sholat dzuhur | valid |
| 3 | Method untuk mengatur tampilan tombol |  | Hasil yang di harapkan dari pengaturan tombol adalah ketika tombol untuk menyimpan data sholat maka tombol akan menghilang artinya ketika kita sudah mencatat sholat maka kita tidak bisa menyimpan data sebanyak 2 kali | valid |
| 4 | Method untuk menset jadwal sholat |  | Hasil yang diharapkan adalah pada menu catatan ibadah waktu sholat yang tampil untuk sholat selanjutnya misal pada waktu 12.30 maka waktu sholat yang selanjutnya ialah ashar | valid |
| 5 | Method untuk menset jumlah waktu sholat dalam detik |  | Hasil yang diharapkan dari kode tersebut ialah waktu yang tampil pada menu jadwal ibadah adalah waktu yang sudah diubah menjadi detik | Valid6 |
| 6 | Method untuk realtime jadwal sholat |  | Kode diatas adalah untuk menyesuaikan jadwal sholat dari API. | valid |

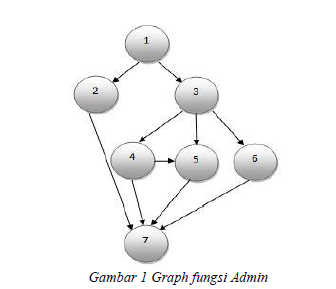
**IMPLEMENTASI PROSES UJI SISTEM INFORMASI ADMISI PASCASARJANA UNIVERSITAS SAM RATULANGI**

1. Identifikasi masalah

Masalah yang terdapat di jurnal ini ialah tester hanya ingin mengetahui bug dan error dari aplikasi ini saja.

1. Hasil analisis

Dari jurnal yang kami analisis terdapat beberapa fungsionalitas yang akan di test oleh tester , yaitu sebagai berikut :



Keterangan tiap node pada graph:

1. Login

2. Tambah data user

3. Lihat data Calon\_Mahasiswa

4. Edit data Calon\_Mahasiswa

5. Cetak data Calon\_Mahasiwa

6. Hapus data Calon\_Mahasiswa

7. Logout

Berdasarkan *flow graph* tersebut terdapat 5 kemungkinan *path* yang ada dengan jumlah edge sebanyak 10. Kemungkinan seluruh *path* adalah:

Jalur Path

1. 1-2-7
2. 1-3-4-7
3. 1-3-4-5-7
4. 1-3-5-7
5. 1-3-6-7

Dengan menggunakan rumus *cyclomatic complexity metric* dapat dihitung nilai kompleksitas dari blok program yang akan diuji. Berikut perhitungannya:

E = 10

N = 7

P = 1

CC= E – N + P

CC = 10 – 7 + 1 = 4

**ANALISIS PERANCANGAN ANIMASI INTERAKTIF PEMBELAJARAN ANATOMI OTAK MANUSIA TINGKAT SEKOLAH MENENGAH PERTAMA**

1. Analisis masalah

Didalam jurnal ini terdapat beberapa masalah yang menjadi topik pembicaraan didalam jurnal tersebut yaitu dalam proses belajar mengajar di MTs N 25 masih mengunakan *one way transfer knowledge system* yaitu dengan menempatkan siswa sebagai objek dan pengajar sebagai subjek. Hal ini membuat siswa cenderung pasif dan kurang menunjukkan tingkat kreatifitas dalam proses pemindahan pengetahuan dari materi yang diberikan oleh pihak pengajar kepada siswa didik.

1. Hasil analisis masalah dengan white box testing
2. 1-2-3-4-5-6-7-8
3. 1-2-3-9-2
4. 1-2-3-4-10-11-2
5. 1-2-3-4-10-12-13-2
6. 1-2-3-4-10-12-14-15-2
7. 1-2-3-4-10-11-12-14-16-17-2
8. 1-2-3-4-5-18-19-20-21-23-24-25-26-27-2
9. 1-2-3-4-5-18-19-20-21-22-24
10. 1-2-3-4-5-18-19-20-21-23-24-25-19
11. 1-2-3-4-5-6-28-2
12. 1-2-3-4-5-6-7-29-2

Analisis *Cyclomatic Complexcity* (CC) adalah: 11 dengan perhitungan mengacu pada (Rumus 1) sebagai berikut.

V(G)= 38-29+2=11

**PERANCANGAN DAN IMPLEMENTASI APLIKASI PELAPORAN PENJUALAN DAN PENGECEKAN STOK BERBASIS ANDROID**

1. Analisis masalah

ditujukan pada perusahaan yang memakai jasa salesmandalam pemasaran produk untuk merancang

dan membuat aplikasi yang berguna bagi perusahaan dalam mengakses data dan informasi.

pengecekan stok barang dan pelaporan penjualan dapat dilakukan secara langsung sehingga monitor

penjualan dapat dilakukan secara lebih cepat tanpa harus menunggu akhir proses pemasaran produk.

1. Hasil analisis masalah dengan white box testing
2. Cari item berhasil - Menampilkan item yang

Dicari

|  |  |  |  |
| --- | --- | --- | --- |
| **Fungsi** | **Hasil yang** | **Hasil** | |
|  | **diharapkan** | **Tampilan** | |
| **void** onClick(View v) { | Cari item berhasil Menampilkan item yang dicari |  | |
| **if** (v == btnCari) { **if** |
| (txtKataKunci.getText().toString().equal |
| s("")) { |
| client.Execute(RequestMethod.*POST*); |
| String response = |
| client.getResponse().trim(); else { |
| AlertDialog.Builder builder = new |
| AlertDialog.Builder(this); |
| builder.setTitle("Cari Item: " + |
| txtKataKunci.getText()); |
| builder.setMessage(response); |
| builder.setPositiveButton("OK", null); |
|  |
|  |
|  |
|  |
|  |  |  | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  | |
|  |
|  |
|  |  |  | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |