# BAB II

# DASAR TEORI

## 2.1 Tinjauan Pustaka

*Tabel 2.1. Tinjauan Pustaka*

| No | Identitas Jurnal | Objek | Metode | Hasil |
|---|---|---|---|---|
| 1. | Disa Arista, Agus Nursikuwagus (2019).<br><br>Pengujian Aplikasi Perhitungan Stok Barang Dengan Metode *Black Box* Pada Cv. Delta Pilar (Cabang Bandung). [online]<br><br>elibrary.unikom.ac.id/id/eprint/689 | Aplikasi Perhitungan Stok Barang Cv. Delta Pilar (Cabang Bandung). | Metode *Black Box* | Hasil dari penelitian ini menunjukkan bahwa aplikasi memiliki kekurangan sebesar 43,4% dengan teknik *BVA+EP* dan 53,5% dengan teknik *DT*. |
| 2 | Taufik Hidayat, Mahmudin Muttaqin (2018).<br><br>Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black *Box Testing* dengan Metode *Equivalence Partitioning* dan *Boundary Value Analysis*.<br><br>Jurnal Teknik | Sistem Informasi Pendaftaran dan Pembayaran Wisuda *Online* | Black *Box Testing* dengan Metode *Equivalence Partitioning* dan *Boundary Value Analysis*. | *Equivalence Partitioning* membahas tentang *testing* dalam aspek validasi inputan dilihat dari *Valid Class*, Pengamatan isi inputan dan akurasi inputan. *Boundary Value Analysis* membahas kepada *testing Black Box* dalam aspek keseluruhan menu dan modul, sehingga dapat diketahui sisi |

| | | | | |
|---|---|---|---|---|
| | Informatika UNIS Vol. 6 No.1, ISSN: 2252-5351 | | | kesalahannya. |
| 3 | Siti Rika Yulistina, Tika Nurmala, Raden Muhamad Agung, Sandi Hidayat, Aries Saifudin. (2020). Penerapan Teknik *Boundary Value Analysis* untuk Pengujian Aplikasi Penjualan Menggunakan Metode *Black Box Testing*.<br><br>Jurnal Informatika Universitas Pamulang ISSN: 2541-1004 Penerbit: Program Studi Teknik Informatika Universitas Pamulang e-ISSN: 2622-4615 Vol. 5, No. 2, Juni 2020 (129-135) | Aplikasi Penjualan | Metode *Black box Testing.* | Hasil dari penerapan metode yang dipakai adalah kualitas dari perangkat lunak sudah sesuai dengan fungsi, serta dapat dimanfaatkan dengan baik oleh pengguna. |
| 4. | Luthfie Auditya Amarul Ma'ruf, Condro Kartiko, Citra Wiguna. (2020) *Black Box Testing Boundary Value Analysis* Pada Aplikasi *Submission System*<br><br>Jurnal Edik Informatika: | *Submission system* pada *website* CENTIVE | *Black Box Testing, Boundary Value Analysis* | Masih banyak kekurangan 5etika memvalidasi data yang akan diinput sehingga dapat menyebabkan data tersimpan dalam database tidak sesuai dengan data yang diharapkan |

| | | | | |
|---|---|---|---|---|
| | Penelitian Bidang Komputer Sains dan Pendidikan Informatika. ejournal.stkip-pgri-umbar.ac.id/index.php/eDikInformatika Vol. 6 No.2 April 2020 | | | |
| 5. | Tri Snadhika Jaya. (2021)<br><br>Pengujian Aplikasi dengan Metode *Blackbox Testing Boundary Value Analysis* (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)<br><br>Jurnal Informatika: Jurnal Pengembangan IT (JPIT), Vol.03, No.02, Januari 2018 | Kantor Digital Politeknik Negeri Lampung | Metode *Blackbox Testing Boundary Value Analysis* | Aplikasi mampu menangani data, baik data normal ataupun data tidak normal dengan persentase keberhasilan 91, 67 %. |

## 2.2 Dasar Teori

Software Testing Pengujian perangkat lunak (*software testing*) adalah aktivitas yang bertujuan untuk mengevaluasi atribut-atribut atau kemampuan sebuah program atau sistem dan penentuan apakah sesuai dengan hasil yang diharapkan atau tidak. Pengujian perangkat lunak spesifiknya adalah proses mengeksekusi suatu program untuk menemukan *bug* dari perangkat lunak. Pengujian yang sukses adalah bila pengujian yang dilakukan berhasil menemukan

suatu kesalahan yang awalnya tidak ditemukan. *Software testing* terdiri dari 8 jenis, yaitu :

1. *Performance Testing*

   *Performance Testing* adalah *integration* dan *usability test* yang menentukan apakah *system* atau *sub system* dapat memenuhi kriteria berbasis waktu seperti *response time* atau *throughput*.

2. *System Testing*

   *System Testing* adalah *integration test* dari *behavior* seluruh sistem atau *independent subsystem*. *System testing* biasanya dilakukan pertama kali oleh pengembang atau personil pengujian untuk memastikan bahwa keseluruhan sistem tidak berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (*user requirement*)

3. *Unit Testing*

   *Unit Testing* adalah proses metode pengujian *individual*, *class*, atau komponen sebelum mereka terintegrasi dengan perangkat lunak lainnya. Tujuannya adalah untuk mengidentifikasi dan memperbaiki kesalahan sebanyak mungkin sebelum modul digabungkan menjadi unit perangkat lunak yang lebih besar.

4. *Integration Testing*

   *Integration Test* proses evaluasi *behavior* dari kelompok *method* atau *class*. Tujuannya adalah untuk mengidentifikasi kesalahan yang tidak dapat dideteksi oleh *unit testing* seperti *Interface Incompability*,

*Parameter Values, Run-Time Exceptions,* dan *Unexpected State Interactions*.

5. *Usability Testing*

   *Usability Test* adalah *test* untuk menentukan apakah *method, class, subsystem*, atau sistem telah memenuhi persyaratan pengguna. Pengujian ini dilakukan untuk mendapatkan *feedback* yang cepat dalam meningkatkan *interface* dan mengkoreksi kesalahan dalam komponen perangkat lunak.

6. *Smoke Testing*

   *Smoke Testing* adalah *system test* yang dilakukan setiap hari atau beberapa kali per minggu. Pengujian ini dilakukan setelah sebuah perangkat lunak telah selesai di *build* untuk memastikan bahwa fungsi penting dari program telah bekerja dengan baik.

7. *Stress Testing*

   *Stress Testing* adalah pengujian yang biasanya dilakukan dalam membuat sebuah *website* untuk mengetahui seberapa kuat *server website* dalam menampung pengunjung.

8. *User Acceptence Test (UAT)*

   *User Acceptance Test* digunakan untuk menentukan apakah sistem yang dikembangkan telah memenuhi kebutuhan pengguna atau belum. Pengujian ini dilakukan setelah rangkaian pengujian seperti *Unit Testing, Integration Testing,* dan *System Testing* selesai dan menggunakan metode *Black Box Testing*.

***Black Box Testing***

Menurut Simanjuntak (2010), menyatakan bahwa *Black-Box Testing* merupakan pengujian perangkat lunak yang merupakan tes fungsionalitas dari aplikasi yang tidak mengacu pada struktur internal atau tidak membutuhkan pengetahuan khusus pada kode program aplikasi dan pengetahuan pemrograman.

Pengujian dengan menggunakan *Black Box* mempunyai beberapa teknik, di antaranya *Equivalence Partitioning, Boundary Value Analysis, Robustness Testing, Behavior Testing* dan *Cause-Effect Relationship Testing* (Safitri dkk, 2018).

Salah satu teknik yang terdapat di dalam *Black Box Testing* mengerjakan proses pada jumlah maksimal dan minimal nilai yang di isikan pada aplikasi disebut *Boundary Value Analysis* (Andriansyah, 2018).

Metode *Black Box testing* terdiri atas beberapa metode, antara lain *Equivalence Partitioning, Boundary Value Analysis, State Transition Testing* dan *Decision Table Testing. Black Box Testing* merupakan metode pengujian perangkat lunak yang digunakan untuk menguji sebuah perangkat lunak tanpa mengetahui struktur internal kode atau program. Dalam pengujiannya, penguji menyadari apa yang harus dilakukan oleh program, tapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

Kelebihan *black box testing* yaitu :

1. Efisien untuk segmen kode besar.
2. Akses kode tidak diperlukan

3. Pemisahan antara perspektif pengguna dan pengembang

Selain memiliki kelebihan, *black box testing* juga memiliki kelemahan, yaitu :

1. Cakupan terbatas karena hanya sebagian kecil dari skenario pengujian yang dilakukan.
2. Pengujian tidak efisien karena keberuntungan tester dari pengetahuan tentang perangkat lunak internal

***Decision Table***

*Decision Table Testing* adalah salah satu metode *Black Box Testing* yang digunakan untuk menguji perilaku sistem dengan beberapa kombinasi input yang berbeda. Metode ini biasa disebut sebagai tabel *Cause-Effect. Decision Table Testing* dilakukan ketika sistem memiliki berbagai variasi nilai input yang berbeda yang tidak bisa diuji melakukan metode *boundary value analysis* ataupun *equivalent partitioning.*