

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Pada penelitian ini terdapat tinjauan pustaka yang membahas tentang penelitian mengenai pengujian aplikasi menggunakan metode *black box testing*. Penelitian tersebut digunakan peneliti sebagai referensi dan menjadi pedoman pendukung penelitian yang dilakukan.

Penelitian dari Dwi Ernawati pada tahun 2023, membahas tentang pengujian pada aplikasi *mobile* Genius HR di PT GIT Solution menggunakan metode *black box testing*. Hasil dari penelitian tersebut menunjukkan bahwa dalam pengujian awal jika dikategorikan dalam bentuk persentase mencakup 75% karena ditemukan *error* atau *bug*. Dan untuk hasil pengujian akhir setelah *error* atau *bug* diperbaiki oleh *developer* telah mencakup 100%. Dengan demikian aplikasi sudah sesuai dengan hasil yang diharapkan.

Penelitian dari Isna Budiarti Utami pada tahun 2023, membahas tentang pengujian pada aplikasi *website* Clandys menggunakan metode *black box testing*. Hasil dari penelitian tersebut menunjukkan bahwa pengujian yang dilakukan memperoleh 22 berhasil dan 1 gagal dari fungsionalitas fitur yang ada. Dengan demikian dapat dikatakan fungsionalitas pada *website* Clandys belum sepenuhnya berjalan dengan baik dan bisa disempurnakan lagi.

Penelitian dari Holis Muchlis Sugiantos dan Asti Herliana pada tahun 2020, membahas tentang pengujian pada aplikasi *mobile* *Reminder* dan *Monitoring* Jadwal Servis Sepeda Motor menggunakan metode *black box testing* setelah proses

pembuatan aplikasi. Hasil dari penelitian tersebut menunjukkan bahwa sistem bebas dari *error*. Sistem yang diuji menunjukkan hasil sesuai dengan yang diharapkan yaitu *user* dapat membuat *reminder* servis sepeda motor yang berguna dalam melakukan servis secara rutin.

Penelitian dari Ahmad Fahrezi, Fahry Noer Salam, Gilang Mahardika Ibrahim, Rifki Rahman Syaiful, dan Aries Saifudin pada tahun 2022, membahas tentang pengujian aplikasi *website* inventori barang PT. AINO Indonesia menggunakan metode *black box testing*. Hasil penelitian tersebut menunjukkan bahwa sistem berjalan sesuai dengan *test case* yang telah dibuat, sehingga aplikasi tersebut dinyatakan dapat digunakan dengan baik. Dan dapat disimpulkan pengujian perangkat lunak sangatlah dibutuhkan demi memastikan dan mengoptimalkan pembangunan sebuah perangkat lunak atau aplikasi sebelum diserahkan kepada *user* untuk digunakan

Penelitian dari Findra Kartika Sari Dewi, Stephanie Pamela Adithama, dan Albert Teonando Suhardi pada tahun 2023, membahas tentang pengujian aplikasi *mobile* pada aplikasi *Doctor to Doctor* menggunakan metode *black box testing*. Hasil penelitian tersebut menunjukkan bahwa dari 16 fungsi yang diuji, rata - rata presentase keberhasilan pengujian adalah 86,89%. Ditemukan 7 *bug* pada 7 fungsi yang diuji yang memerlukan perbaikan oleh pengembang agar kualitas aplikasi *Doctor to Doctor* menjadi lebih baik. Berdasarkan kriteria kelayakan menurut Arikuntoro, Aplikasi *Doctor to Doctor* telah dikategorikan sebagai sangat baik.

Berikut merupakan tabel tinjauan pustaka yang digunakan sebagai referensi dan menjadi pedoman pendukung penelitian yang dilakukan. Dapat dilihat pada tabel 2.1.

**Tabel 2.1 Tinjauan Pustaka**

No.	Nama (Tahun)	Topik	Metode/ Pendekatan	Hasil
1.	Dwi Ernawati (2023)	Pengujian pada aplikasi mobile Genius HR di PT GIT Solution menggunakan metode black box testing	<i>Black Box Testing</i>	Hasil dari penelitian tersebut menunjukkan bahwa dalam pengujian awal jika dikategorikan dalam bentuk persentase mencakup 75% karena ditemukan error atau bug. Dan untuk hasil pengujian akhir setelah error atau bug diperbaiki oleh developer telah mencakup 100%. Dengan demikian aplikasi sudah sesuai dengan hasil yang diharapkan.
2.	Isna Budiarti Utami (2023)	Pengujian pada aplikasi website Clandys menggunakan metode black box testing	<i>Black Box Testing</i>	Hasil dari penelitian tersebut menunjukkan bahwa pengujian yang dilakukan memperoleh 22 berhasil dan 1 gagal dari fungsionalitas fitur yang ada. Dengan demikian dapat dikatakan fungsionalitas pada website Clandys belum sepenuhnya berjalan dengan baik dan bisa disempurnakan lagi.
3.	Holis Muchlis Sugiantos dan Asti Herliana (2020)	Pengujian pada aplikasi mobile Reminder dan Monitoring Jadwal Servis Sepeda Motor menggunakan metode black box testing setelah proses pembuatan aplikasi	<i>Black Box Testing</i>	Hasil dari penelitian tersebut menunjukkan bahwa sistem bebas dari error. Sistem yang diuji menunjukkan hasil sesuai dengan yang diharapkan yaitu user dapat membuat reminder servis sepeda motor yang berguna dalam melakukan servis secara rutin.
4.	Ahmad Fahrezi, Fahry Noer Salam, Gilang Mahardika Ibrahim, Rifki Rahman Syaiful, dan Aries Saifudin (2022)	Pengujian aplikasi website inventori barang PT. AINO Indonesia menggunakan metode black box testing	<i>Black Box Testing</i>	Hasil penelitian tersebut menunjukkan bahwa sistem berjalan sesuai dengan test case yang telah dibuat, sehingga aplikasi tersebut dinyatakan dapat digunakan dengan baik. Dan dapat disimpulkan pengujian perangkat lunak sangatlah dibutuhkan demi memastikan

**Tabel 2.1 Tinjauan Pustaka (Lanjutan)**

No.	Nama (Tahun)	Topik	Metode/ Pendekatan	Hasil
5.	Findra Kartika Sari Dewi, Stephanie Pamela Adithama, dan Albert Teonando Suhardi (2023)	Pengujian aplikasi mobile pada aplikasi Doctor to Doctor menggunakan metode black box testing	<i>Black Box Testing</i>	Hasil penelitian tersebut menunjukkan bahwa dari 16 fungsi yang diuji, rata - rata presentase keberhasilan pengujian adalah 86,89%. Ditemukan 7 bug pada 7 fungsi yang diuji yang memerlukan perbaikan oleh pengembang agar kualitas aplikasi Doctor to Doctor menjadi lebih baik. Berdasarkan kriteria kelayakan menurut Arikuntoro, Aplikasi Doctor to Doctor telah dikategorikan sebagai sangat baik.

## 2.2 Dasar Teori

### 2.2.1 Software Testing

*Software testing* merupakan sebuah proses pengujian perangkat lunak dengan tujuan untuk menemukan kegagalan dan mengungkapkan informasi tentang kualitas perangkat lunak. Menurut Boris Beizer (2008) *software testing* adalah proses identifikasi kesalahan, kekurangan, atau ketidaksesuaian pada perangkat lunak yang sedang diuji. Berikut ini adalah penjelasan lebih detail pandangan Boris Beizer mengenai tujuan software testing:

1. Mengidentifikasi kesalahan dan kegagalan dalam perangkat lunak.
2. Mencari kekurangan dalam perangkat lunak yang dapat mempengaruhi kinerja atau kualitasnya.
3. Memverifikasi apakah perangkat lunak memenuhi persyaratan yang ditetapkan dan berfungsi sesuai dengan yang diharapkan.

Dalam *software testing* terdapat beberapa metode pengujian diantaranya adalah *White Box testing*, *Grey Tetsting*, dan *Black box Testing*. *White box testing* merupakan metode pengujian perangkat lunak yang melibatkan pemahaman mengenai struktur internal perangkat lunak. Penguji harus memiliki pengetahuan tentang bagaimana perangkat lunak diimplementasikan, termasuk kode pemrograman, algoritma, dan struktur data. Informasi tersebut digunakan penguji untuk merancang dan menjalankan pengujian lebih terperinci.

*Black box testing* merupakan metode pengujian perangkat lunak dengan melakukan pengujian tanpa memperhatikan implementasi internalnya. Dalam *black box testing* penguji tidak perlu memiliki pengetahuan terperinci tentang bagaimana perangkat lunak diimplementasikan, tetapi fokus pada *input* yang diberikan dan *output* yang dihasilkan oleh perangkat lunak.

*Grey box testing* merupakan metode pengujian kombinasi dari *white box testing* dan *black box testing*. Penguji dalam metode *grey box testing* perlu memiliki sebagian pengetahuan tentang implementasi internal perangkat lunak, sehingga dapat mempengaruhi desain pengujian dengan memperhatikan aspek – aspek internal tertentu namun tetap fokus pada pengujian berdasarkan proses input dan hasil output dari perangkat lunak.

Metode – metode tersebut dapat digunakan secara terpisah atau dikombinasikan sesuai dengan kebutuhan dan tujuan pengujian. Tujuan utama *software testing* adalah memastikan perangkat lunak berfungsi dengan baik, memenuhi persyaratan, and memberikan pengalaman yang baik bagi pengguna.

### 2.2.2 Black Box Testing

*Black box testing* atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan dengan menguji keluaran yang dihasilkan berdasarkan nilai masukan yang diberikan tanpa mengetahui kode program yang digunakan. Pengujian menggunakan *Black Box Testing* didasarkan pada detail aplikasi seperti fungsi-fungsi yang ada pada aplikasi, tampilan aplikasi, dan kesesuaian alur fungsi.

Menurut Boris Beizer (2008) *black box testing* adalah pendekatan pengujian perangkat lunak di mana pengujian dilakukan berdasarkan perilaku dan persyaratan eksternal perangkat lunak, tanpa memperhatikan struktur internal atau rincian implementasi. Pendekatan ini melibatkan pengujian fungsi-fungsi perangkat lunak dengan memberikan input dan mengamati output yang dihasilkan. Tujuan dari *black box testing* adalah untuk menguji apakah perangkat lunak berperilaku sesuai dengan persyaratan yang telah ditetapkan tanpa mengetahui atau memahami bagaimana perangkat lunak bekerja didalamnya.

### 2.2.3 Pengujian Manual

Menurut Damar Mustiko Aji (2018) pengujian manual adalah aktivitas untuk mengetahui *bug*, *error*, dan *defect* pada suatu perangkat lunak. Seseorang yang melakukan aktivitas tersebut biasa disebut *tester*, atau dalam dunia pekerjaan bidang IT dikenal dengan *Quality Assurance (QA)*, *Software Tester*, dan *QA Engineer*. Dalam metode ini seorang penguji atau *tester* berperan sebagai *end user* atau pengguna perangkat lunak tersebut. Pengujian dilakukan secara manual dengan mengeksekusi kasus uji atau *test case* tanpa menggunakan *software* ataupun *tools*. Tujuan utama pengujian manual adalah untuk memastikan bahwa aplikasi

yang diuji bebas cacat dan aplikasi berkerja sesuai dengan ketentuan atau spesifikasi yang telah ditentukan. Terdapat kelebihan yang hanya bisa dilakukan oleh pengujian manual seperti mendapatkan *visual feedback* tampilan *UI/UX* aplikasi dan mendapatkan feedback dari orang atau *end user* secara langsung.

#### **2.2.4 Aplikasi Pengingat Catatan Dengan Manajemen Tim**

Aplikasi Pengingat Catatan Dengan Manajemen Tim merupakan sebuah aplikasi seperti asisten pribadi, dengan cara mengingatkan catatan yang sudah dibuat. Pengingat yang akan terjadi dapat disesuaikan dengan keinginan pengguna. Pengingat tersebut akan berbentuk sebuah notifikasi dengan alarm. Untuk mematikan alarm dapat dilakukan dengan menekan notifikasi atau menekan tombol *stop* pada halaman *fullscreen* notifikasi alarm. Terdapat dua jenis pengingat, yaitu pengingat sebelum tanggal acara dan pengingat ketika tanggal acara. Pengingat akan muncul sesuai dengan tanggal dan waktu yang sudah dimasukkan. Notifikasi alarm aplikasi ini akan tetap menyala ketika ponsel berada dalam mode Jangan Ganggu atau *Do Not Distrub*.

Aplikasi ini dapat digunakan secara pribadi dan juga dapat melakukan kerja sama dengan orang lain ketika memiliki ketergantungan pekerjaan atau bisa disebut dengan kolaborasi atau tim. Untuk melakukan kolaborasi dapat dilakukan dengan mengundang menggunakan email. Email yang dimasukkan harus dipastikan sudah memiliki akun atau sudah terdaftar. Undangan tersebut nantinya akan muncul dan bisa dikonfirmasi melalui Beranda aplikasi atau bisa juga di menu Undangan.

Dalam catatan kolaborasi orang yang mengundang disebut dengan pemilik sedangkan orang yang menerima undangan disebut dengan member. Pemilik dapat mengubah catatan, menambah member, menghapus member, dan menghapus catatan dengan syarat wajib mengisikan pesan yang nantinya akan dikirimkan ke member melalui notifikasi. Sedangkan member hanya bisa mengupload file bukti atau melampirkan tugas yang sudah selesai dikerjakan. Ketika member menyelesaikan tugas atau sudah mengunggah file, akan terlihat dalam status progres yang hanya dapat dilihat oleh pemilik catatan. Untuk file yang diunggah oleh member dapat dilihat dan didownload oleh pemilik, sedangkan member hanya bisa melihat saja.

Selain catatan pribadi dan kolaborasi, terdapat fitur tim yang dapat digunakan untuk melakukan manajemen dalam sebuah proyek atau manajemen progress suatu organisasi. Dalam aplikasi ini, tim dimuat dalam sebuah halaman dashboard yang menampilkan list yang dapat diisi catatan seperti kanban. Semua member tim dapat menambah list, mengubah list, menghapus list, menambah catatan, mengubah catatan, dan menghapus catatan. Kemudian untuk menambah member tim, menghapus member tim, dan mengubah detail tim, hanya bisa dilakukan oleh Admin tim. Sama seperti catatan kolaborasi, untuk bergabung ke tim dapat diundang menggunakan email yang sudah terdaftar oleh Admin. Undangan tersebut nantinya akan muncul di Beranda aplikasi atau bisa juga dibuka di menu Undangan.

Dalam aplikasi ini juga menggunakan sistem poin dan kuota catatan. Setiap berhasil daftar, akun secara *default* akan mendapatkan 300 poin dan 3 kuota catatan.



Kuota catatan dapat dibeli atau ditukarkan dengan poin yang dimiliki. Untuk saat ini, cara untuk mendapatkan poin dapat dilakukan dengan menyelesaikan catatan pribadi dan catatan kolaborasi.

### 2.2.5 Test Case

Menurut Peni Kurniawati (2018) *test case* atau bisa disebut dengan uji kasus adalah suatu rancangan atau rangkaian mengenai tindakan yang dilakukan oleh *user*, dalam proses pengujian tindakan tersebut dilakukan oleh Quality Assurance atau tester untuk melakukan verifikasi terhadap fitur atau fungsi tertentu dari sebuah perangkat lunak.

Pembuatan *test case* bertujuan untuk memastikan bahwa suatu sistem dapat dijalankan dengan baik sesuai dengan kebutuhan yang sudah ditentukan serta mampu memberikan respon ketika terdapat suatu masukan yang tidak *valid*. *Test case* memiliki beberapa komponen seperti *test case id*, *test case description*, *precondition*, *test step*, *expected result*, *actual result*, serta *status*. Dalam penelitian ini komponen yang biasa digunakan dalam sebuah dokumen *test case* adalah sebagai berikut:

1. *Test Case ID*: Identifikasi dari kasus uji.
2. *Type*: Jenis kasus uji, positive atau negative.
3. *Test Case Title*: Deskripsi singkat kasus pengujian.
4. *Pre-Condition*: Prasyarat yang dilakukan sebelum melakukan pengujian.
5. *Steps Description*: Langkah – langkah pengujian.
6. *Expected Result*: Hasil yang diharapkan.

7. *Actual Result*: Hasil setelah melakukan langkah – langkah pengujian.

8. *Status*: Status uji, Pass/ Fail dari eksekusi test case.

*Test case* bertindak sebagai titik awal dalam pelaksanaan pengujian sebuah sistem. Dari *test case* ini biasanya diketahui apakah fitur sistem berjalan normal atau tidak.