



Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing

Dede Wintana¹, Denny Pribadi², Moh Yusuf Nurhadi³

^{1,2}Ilmu Komputer Kampus Kota Sukabumi, Teknik dan Informatika, Universitas Bina Sarana Informatika

Jl.Sukakarya, Warudoyong Kota Sukabumi

Dede.dwe@bsi.ac.id, Denny.dpi@bsi.ac.id, m.yusupnh06@gmail.com

Pada penelitian ini akan dijelaskan perbandingan efektivitas pengujian black-box dengan pengujian white-box. Peneliti akan membandingkan seberapa efisien pengujian kotak putih dan pengujian kotak hitam. Dalam penggunaannya mana yang lebih baik dan mana yang lebih buruk. Dalam penelitian ini, kami akan membandingkan dengan data seberapa efisien pengujian tersebut. Berdasarkan pengujian yang dilakukan dengan white-box testing, pengujian memerlukan persiapan teknis dan awal yang ditentukan sebagai pengujian otomatis diperlukan untuk membuat skrip pengujian otomatis agar pengujian white-box bekerja seperti yang diharapkan, Jadi pengujian white-box membutuhkan awal yang lebih lama uji persiapan daripada pengujian kotak hitam yang membutuhkan waktu untuk penggunaan awal. Tetapi pengujian white-box lebih cepat daripada pengujian black-box yang hanya menjalankan skrip dalam satu klik dan hasilnya akan muncul. Pada grafik yang dihasilkan pengujian white-box dan perbandingan pengujian black-box. hasil penelitian menyimpulkan bahwa pengujian black box lebih baik daripada pengujian white-box apabila pengujian dengan frekuensi yang lebih sedikit, berdasarkan hasil yang diperoleh dapat disimpulkan bahwa pengujian white-box lebih baik daripada pengujian back-box untuk pengujian dengan frekuensi yang lebih banyak. . . Data tersebut juga menunjukkan kepada kita bahwa white-box testing dan back-box testing memiliki kelebihan dan kekurangannya masing-masing, tergantung kebutuhan mana yang dibutuhkan.

Kata kunci: manual testing, white-box testing, testing, perbandingan testing, perbandingan efisiensi testing.

Abstract - In this study, we will describe the comparison of the effectiveness of black-box testing with white-box testing. Researchers will compare how efficient white-box testing and black-box testing are. In its use which one is better and which one is worse. In this study, we will compare with the data how efficient the testing is. Based on the tests carried out by white-box testing, testing requires technical and initial preparation which is determined as the automated testers needed to create automated test scripts so that white-box testing works as expected, So white-box testing requires a longer initial preparation test than black -box testing which takes time for initial use. But white-box testing is faster than black-box testing which just runs the script in one click and the results will appear. In the resulting graph white-box testing and comparison of black-box testing, the results of the study conclude that black box testing is better than white-box testing when testing with less frequency, based on the results obtained, it can be concluded that white-box testing is better than back-box testing for testing with more frequency. . The data also shows us that white-box testing and back-box testing have their respective advantages and disadvantages, depending on which needs are needed.

Keywords: manual testing, white-box testing, testing, testing comparison, efficiency testing comparison.

* Korepondensi.

Alamat E-mail : jurnal.larik@bsi.ac.id

Diterima 30 April 2022; Direvisi 20 Juni 2022; Diterima 10 Juli 2022

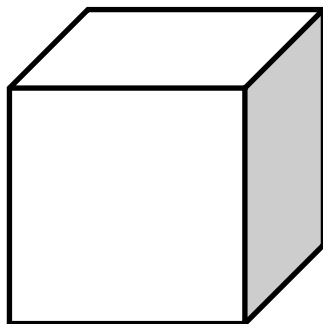
I. PENDAHULUAN

Di era Industri 4.0 semuanya berjalan lebih cepat, yang membuat waktu pengembangan lebih kompetitif. Periode pengembangan dan periode inovasi perlu dipersingkat. Kemampuan inovasi yang tinggi menjadi faktor penentu keberhasilan bagi banyak perusahaan [1]. Dalam makalah ini, kita akan membahas tentang seberapa efektif black-box testing membandingkan manual testing dengan.

membandingkan langkah demi langkah yang diambil. Kami akan melihat langkah berapa lama, berapa lama waktu yang dibutuhkan untuk membuat black-box testing berfungsi. Dan kita akan melihat testing tentang menggunakan manual testing seberapa efektif itu.

Jadi di sini penulis akan membahas lebih efektif manakah antara white-box testing jika dibandingkan dengan manual testing. Apa itu manual testing? Mengapa white-box testing? Dan kenapa bukan black-box testing? Berikut akan penulis bahas secara terperinci kenapa penulis white-box testing dengan manual testing.

Jadi mengapa testing dengan white-box? White-box testing terutama berfokus pada logika internal dan struktur kode. Desain uji kasus yang menguji fungsi internal perangkat lunak dari perspektif pengembang. White-box dilakukan ketika programmer memiliki teknik pengetahuan penuh tentang struktur program. Dengan teknik ini, Anda dapat menguji setiap cabang dan keputusan dalam program. [2].

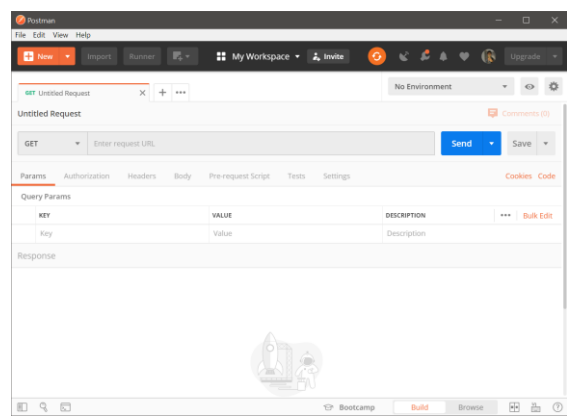


Gambar 1. White-box

Saat kami tidak membahas tentang black-box testing. Tetapi beberapa orang masih mempertanyakan tentang black-box testing. Black-box testing adalah jenis testing yang memperlakukan perangkat lunak tanpa kinerja internal yang

diketahui sehingga penguji melihat perangkat lunak seperti black-box yang tidak penting untuk melihat isinya, tetapi cukup tunduk pada proses testing luarnya. [3].

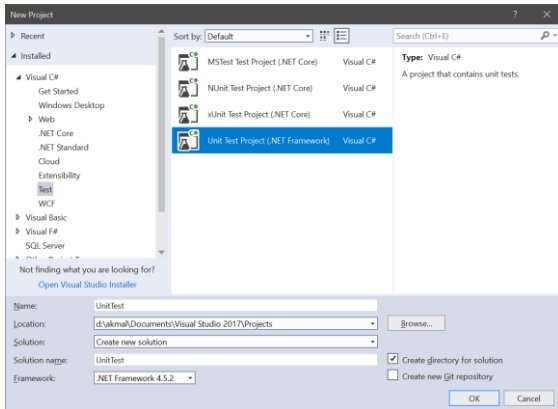
Kemudian peneliti menyebut manual testing sebagai tes yang dilakukan oleh manusia, menggunakan cara klik mouse dan keyboard. Tanpa otomatisasi tunggal hanya menggunakan skrip testing. Lakukan penuh secara manual hingga selesai. Metode testing ini juga dikenal sebagai metode testing paling tidak efisien yang pernah ada.



Gambar 2. Screenshot Postman untuk Keperluan Manual Testing

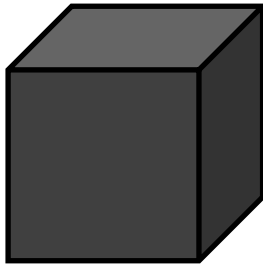
Pada manual testing ini penulis menggunakan Postman untuk menguji enkripsi yang berjalan. Aplikasi Postman ini adalah Sebuah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat. Fungsi Postman adalah untuk pengecekan web service. Postman dapat menampilkan hasil dari HTTP request yang kompleks sekalipun dengan cepat [4].

Salah satu bagian terpenting dari pengembangan aplikasi adalah testingnya. Proses testing selalu amat penting dan krusial dalam siklus pengembangan perangkat lunak, itulah sebabnya perlu perhatian khusus dalam melakukan testing dari pengembangan perangkat lunak tersebut. Prosedur testing yang tepat secara signifikan meningkatkan tingkat kualitas produk yang dikembangkan [5].



Gambar 3. Salah Satu Fitur Unit Test pada Visual Studio 2017

Penelitian ini menggunakan white-box testing yang di dalamnya menggunakan unit test. Tujuan testing dengan unit test ini adalah untuk mencapai cakupan struktural yang tinggi dari kode yang sedang diuji. Pencapaian dari kode berorientasi objek membutuhkan pemanggilan metode sekuensial yang diinginkan dan bermutasi objek. Urutan ini membantu menghasilkan target objek seperti argumen atau status objek penerima (singkatnya sebagai status target) dari metode yang sedang diuji. Pembuatan sekuens otomatis untuk mencapai target sering kali menantang karena ruang pencarian sekuensial yang mungkin besar [6].

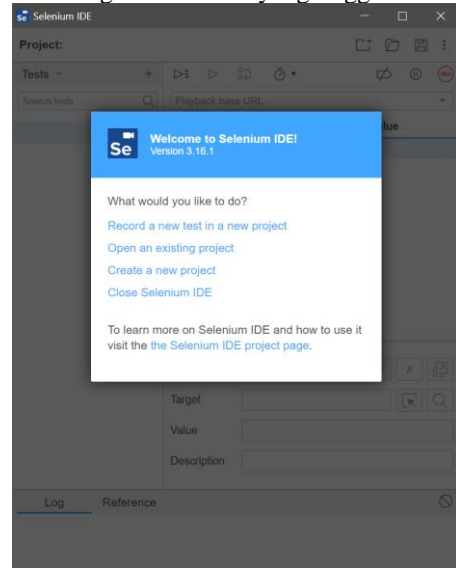


Gambar 4. Black-box

Sebagai black-box testing biasanya memicu beberapa hal. Pada aplikasi web, black-box testing kadang terdeteksi sebagai serangan brute force atau serangan bot. Entah bagaimana aplikasi memblokir black-box testing dan mengharuskan pengguna untuk memasukkan captcha. Jadi, apakah peneliti tidak ingin menggunakan black-box testing.

karena mungkin tidak memiliki kemampuan untuk mengembangkan aplikasi dengan fitur

keamanan yang kadang dibutuhkan oleh beberapa perusahaan dengan keamanan yang tinggi.



Gambar 5. Selenium IDE Salah Satu Contoh Tools Black-Box Testing

White-box testing menguji lebih dalam dari black-box testing jadi mengapa peneliti memilih metode ini dan menggunakan uji unit. Karena ini menguji lebih dalam dan hasilnya dapat memeriksa lebih kompleks dengan pemrograman dan proses yang berjalan. Menguji lebih dalam juga berarti bisa lebih memeriksa seberapa efisien pemrograman itu sendiri dan memeriksa kesalahan dari dalam. Bahkan antarmuka pengguna grafis atau disebut GUI tidak menunjukkan kesalahan.

```

namespace EncryptDecrypt2WayTest
{
    [TestClass]
    [0 references | Almal Digentara, 41 days ago | 1 author, 1 change]
    public class EncryptionTest
    {
        [TestMethod]
        [0 references | Almal Digentara, 41 days ago | 1 author, 1 change]
        public void EncryptDecryptClassTest()
        {
            string PlainText = KeyGenerator.GetUniqueKey(10);
            Thread.Sleep(10);
            string Key = KeyGenerator.GetUniqueKey(10);
            Thread.Sleep(10);
            string Salt = KeyGenerator.GetUniqueKey(10);
            Thread.Sleep(10);

            string Encrypted_string = AesEncryption.Encrypt(PlainText, Key, Salt);
            string Decrypted_string = AesEncryption.Decrypt(Encrypted_string, Key, Salt);

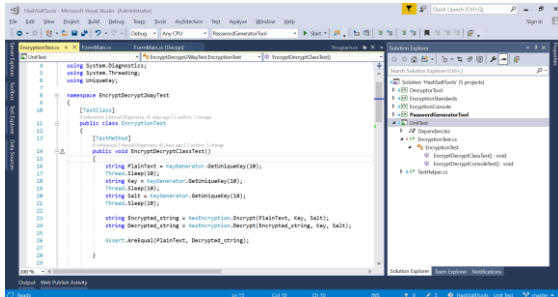
            Assert.AreEqual(PlainText, Decrypted_string);
        }
    }
}

```

Gambar 6. Salah Satu Contoh Scripting pada Unit Test White-Box Testing

Apakah itu GUI? GUI atau singkatan dari Graphical User Interface yang memungkinkan pengguna untuk berinteraksi dengan perangkat keras pada komputer serta dapat memudahkan dalam

mengoperasikan sebuah sistem operasi (user friendly). GUI merupakan suatu sarana suatu penghubung antara pengguna (user) dengan apa yang sudah digunakan tersebut [7].



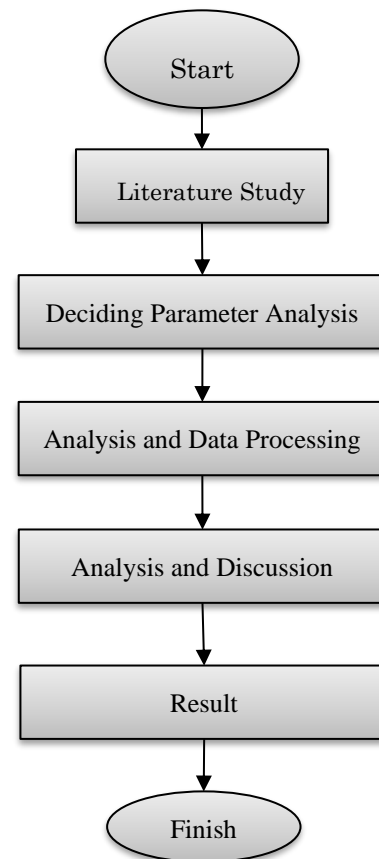
Gambar 7. Salah Satu Contoh GUI sebagai IDE development

Metode automated testing ini adalah metode yang lebih baik. Dan juga tester membutuhkan keterampilan khusus. Dan posisi tester itu sendiri tidak disebut hanya tester. Ini disebut automated testing. Tugas testing otomatis adalah membuat skrip testing otomatis. Biasanya unit test, black-box testing, white-box testing dan lainnya.

Beberapa tester juga diharuskan untuk membuat sendiri test case sehingga mereka membuat test case kemudian membuat skrip testing otomatis, kemudian mengeksekusi skrip testing. Dibandingkan dengan tester biasa. Penguji reguler hanya menguji skrip testing apa yang diperlukan untuk menguji. Tetapi beberapa tester bisa juga membuat test case sendiri.

Penguji aplikasi perangkat lunak dapat menggunakan perangkat komputasi 130 dan browser yang didukung 135 untuk mengakses situs web pengembang dan penguji 140 untuk mengakses antarmuka domain cloud 155. Akses alamat jaringan 140 dan 145 merupakan metode secure login langsung yang dapat diterapkan untuk berdiri aplikasi mandiri berbasis web, misalnya aplikasi berbasis Java, Perl, dan Python yang digunakan untuk mengakses antarmuka domain cloud 155, tanpa perlu browser [8].

Meneliti mengambil penelitian ini dengan cara ini:



Gambar 8. Alur Penelitian

II. METODE PENELITIAN

Penelitian ini akan dimulai dalam studi literatur yang membuat kita membahas tentang teori yang digunakan dalam penelitian lain sebelumnya. Dan menentukan analisis parameter yang digunakan dalam analisis data.

Pada penelitian ini, peneliti menggunakan beberapa parameter. Skenario parameter yang digunakan diantaranya:.

1. Initial preparation.
2. Man hours testing.
3. Effectivity for multiple time.

Penelitian ini berarti membandingkan cara white-box testing yang efektif dibandingkan dengan manual testing. White-box mungkin testing bisa efektif dan kuat yang membuat testing lebih kuat. Ada beberapa keuntungan dari white-box di bawah ini [2].

1. Logic Error

Gunakan sintaks 'jika' dan sintaks pengulangan. Langkah selanjutnya dari metode white-box testing ini adalah mencari dan mendeteksi semua kondisi yang diyakini tidak sesuai dan dicari ketika proses iterasi berakhir.

2. Assumption mismatch

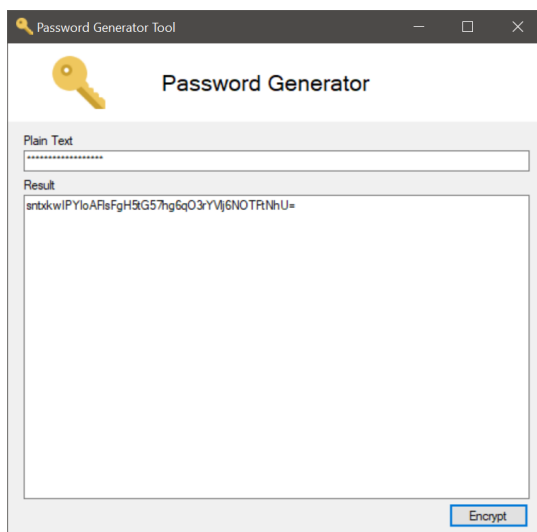
Peragakan dan pantau beberapa asumsi yang diyakini tidak sesuai dengan yang diharapkan atau direalisasikan, untuk dianalisis lebih lanjut dan kemudian diperbaiki.

3. Typing Errors

Mendeteksi dan mencari bahasa pemrograman yang dianggap case-sensitive.

Jadi peneliti mendapatkan data dari aplikasi enkripsi yang memiliki beberapa fitur seperti time-based encryption, decryption, juga dengan static encryption dan decryption yang diuji setiap kali aplikasi mengalami perubahan untuk memastikan tidak ada kesalahan karena aplikasi ini digunakan sebagai tools untuk manual testing aplikasi lain.

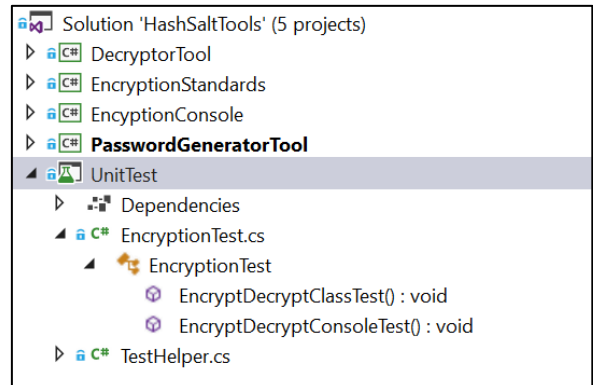
Peneliti menggunakan aplikasi dari suatu PT. XYZ dengan nama disamarkan, hash dan salt key yang disamarkan juga untuk melakukan penelitian ini. Aplikasi yang digunakan merupakan aplikasi password generator atau bisa disebut juga password encryptor yang digunakan untuk menyimpan password, demi menghindari penyimpanan password dalam bentuk plain text, dengan cara melakukan enkripsi sebelum menyimpan password ke dalam text yang tidak terproteksi.



Gambar 9. Aplikasi Encryption

Jadi peneliti menghitung berapa banyak waktu yang diperlukan untuk membuat white-box testing dan memasukkan data. Peneliti juga menghitung berapa banyak waktu yang dibutuhkan untuk persiapan awal dan waktu pengujian sebagai persiapan untuk membuat skrip white-box testing.

dan berapa banyak waktu yang diperlukan untuk menjalankan skrip pengujian.



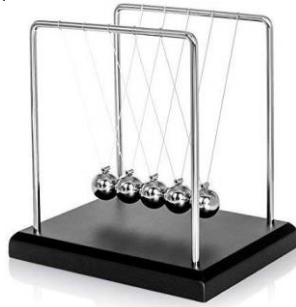
Gambar 10. Solution Explorer dari Unit Test Project Terkait

Sebelum itu peneliti menggunakan manual testing dan menghitungnya berapa banyak waktu yang dibutuhkan untuk manual testing itu sendiri dan menghitungnya berapa banyak waktu yang diulang. Dan menjadikannya sebagai data untuk penggunaan grafik.

Juga diingat, kinerja perangkat komputasi frontend dapat dibatasi oleh faktor-faktor yang tidak tergantung secara signifikan pada perangkat keras pemrosesan, seperti konektivitas jaringan dan aplikasi antarmuka pengguna untuk berinteraksi dengan aplikasi perangkat lunak. Aplikasi perangkat lunak antarmuka pengguna biasanya mendukung platform client-side dan server-side scripting [8].

Jadi hasil pengujian dapat mengambil hasil yang berbeda karena komputer yang diambil oleh peneliti memiliki spesifikasi yang berbeda dari pemeriksa ulang memeriksa kembali data dari hasil yang peneliti berikan. Juga dengan keterampilan berbeda dari automated testing dan pengujian reguler dapat memengaruhi data white-box testing dan manual testing karena keterampilan yang dikenal juga memengaruhi kecepatan kerja. Yang juga diketahui mempengaruhi sistem ekonomi perusahaan. Sebagai dampak ekonomi perusahaan juga mempengaruhi kemampuan membeli komputer

dengan spesifikasi lebih tinggi yang membuat pengujian lebih cepat. Jadi ini lebih seperti pendulum. Efek yang bergerak satu sama lain.



Gambar 11. Pendulum

White-box testing (juga dikenal sebagai clear box testing, glass box testing, transparent box testing, dan structural testing) adalah metode pengujian software yang menguji struktur internal atau cara kerja suatu aplikasi, yang bertentangan dengan fungsinya (yaitu black-box testing). Dalam white-box testing, perspektif internal sistem, serta keterampilan pemrograman, digunakan untuk merancang kasus pengujian.

Tester memilih input untuk menjalankan jalur melalui kode dan menentukan output yang diharapkan. Ini analog dengan pengujian node dalam suatu rangkaian, mis. In-circuit testing (ICT). White-box testing dapat diterapkan pada tingkat unit, integrasi, dan sistem dari proses pengujian perangkat lunak. Meskipun pengujian tradisional cenderung menganggap white-box testing dilakukan di tingkat unit, ini digunakan untuk integrasi dan pengujian sistem yang lebih sering saat ini. Ini dapat menguji jalur dalam suatu unit, jalur antara unit selama integrasi, dan antara subsistem selama pengujian tingkat sistem. Meskipun metode desain uji ini dapat mengungkap banyak kesalahan atau masalah, metode ini berpotensi kehilangan bagian spesifikasi yang tidak diterapkan atau persyaratan yang hilang.

HASIL DAN PEMBAHASAN

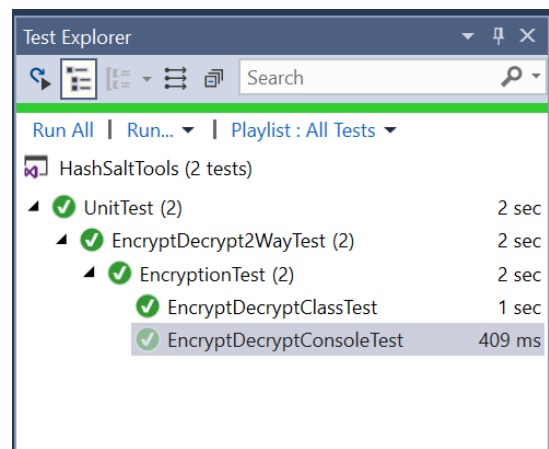
Berdasarkan hasil peneliti, pengujian peneliti mendapatkan hasil seperti di bawah ini:

Tabel 1. Durasi testing dalam hitungan detik

	<i>Initial preparation (seconds)</i>	<i>Man hours testing time (seconds)</i>
White Box Testing	1114	2
Black-box Testing	30	15

<i>White Box Testing</i>	1114	2
<i>Black-box Testing</i>	30	15

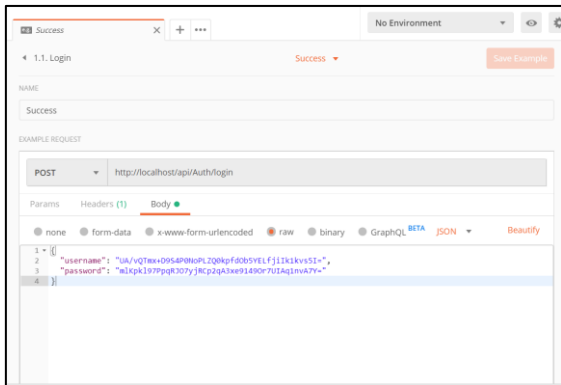
Itu berarti persiapan awal sebelum white-box testing seperti pengkodean white-box testing dapat berlangsung hingga 1114 detik yang berarti 18:34 menit diperlukan untuk mengambil pada persiapan white-box testing pertama kalinya, waktu ini diambil dari date modified dimana project unit test pertama dibuat, lalu penulis membandingkannya ke date modified akhir dimana project unit test diselesaikan hingga mendapatkan 18 menit 34 detik tersebut yang lalu dikonversikan ke dalam satuan detik menjadi 1114 detik. Lalu penulis melakukan start timer sebelum menjalankan unit test dan stop timer ketika unit test berhasil. Maka dari hasil tersebut penulis mendapatkan waktu pengujian hanya membutuhkan waktu sekitar 2 detik.



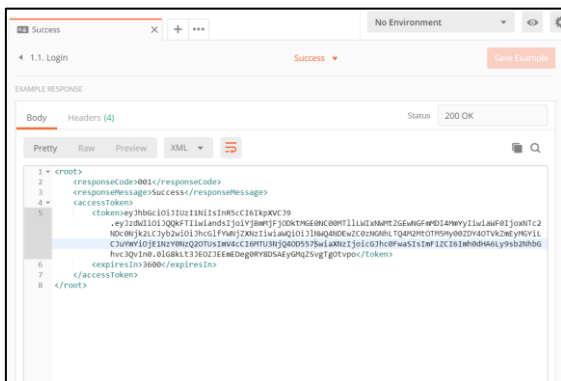
Gambar 12. Hasil White-box Testing

Pengujian manual hanya inisialisasi waktu singkat, ini hanya membutuhkan waktu 30 detik, hal ini lambat karena kinerja komputer membutuhkan waktu. Waktu yang dibutuhkan ini adalah waktu yang diperlukan untuk menyiapkan aplikasi dan membuka aplikasi tersebut. Kemudian pengujian selanjutnya diperlukan sekitar 15 detik untuk setiap pengujian, tergolong cukup lambat dikarenakan diperlukan ketelitian dari tester untuk mencocokkannya ataupun menggunakan aplikasi yang sudah ada untuk menguji enkripsi. Enkripsi

tergolong berhasil apabila aplikasi bisa melakukan decryption dengan baik dan benar tanpa masalah sedikit pun.



Gambar 13. Pengtesan Secara Langsung Menggunakan Postman dengan Enkripsi yang Dihasilkan



Gambar 14. Contoh Testing yang Berhasil Menggunakan Postman Menghasilkan Return Success

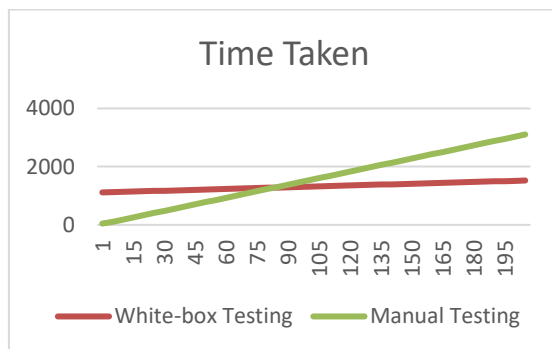
Sebagaimana peneliti menghitung lebih banyak kali Anda menguji menggunakan manual testing yang harganya lebih banyak kali daripada white-box testing. Data ditampilkan di bawah ini.

Tabel 2. Durasi testing dalam hitungan detik

Repetition	White-box Testing	Manual Testing
1	1116	45
2	1118	60
3	1120	75
4	1122	90
5	1124	105
6	1126	120
7	1128	135
8	1130	150
9	1132	165
10	1134	180
11	1136	195
12	1138	210
13	1140	225
14	1142	240
15	1144	255
16	1146	270
17	1148	285
18	1150	300
19	1152	315
20	1154	330
...
...
200	1514	3030
201	1516	3045
202	1518	3060
203	1520	3075
204	1522	3090
205	1524	3105

Pengulangan disebut berapa kali Anda menguji menggunakan metode yang ditentukan. Pengujian White Box mewakili berapa lama dalam detik waktu yang dibutuhkan. Pengujian Manual menunjukkan berapa lama waktu yang diambil dalam detik.

Sebagai peneliti menghitung grafik ini akan menggambarkan seperti yang ditampilkan di bawah ini.



Gambar 15. Grafik Time Taken

Gambar ini seperti yang ditunjukkan di atas, Anda dapat melihat manual testing tidak efisien. Pada manual testing pertama sangat cepat. Mengapa? Karena manual testing tidak memerlukan teknis khusus, cukup gunakan aplikasi kemudian jalankan. Kemudian periksa apakah aplikasi berjalan dengan lancar seperti yang diharapkan atau tidak.

Tetapi dalam white-box testing, pengujian diperlukan persiapan teknis dan awal yang ditentukan sebagai pengujian otomatis yang diperlukan untuk membuat skrip pengujian otomatis agar white-box testing berfungsi seperti yang diharapkan.

Jadi white-box testing memerlukan pengujian persiapan awal lebih lama dari manual testing yang membutuhkan waktu untuk penggunaan awal. Tetapi white-box testing lebih cepat daripada manual testing yang hanya menjalankan skrip dalam satu klik dan hasilnya akan muncul.

III. KESIMPULAN DAN SARAN

Dari penelitian tersebut, peneliti menganggap dari data hasil penelitian tersebut, hasil penelitian menyimpulkan bahwa manual testing lebih baik daripada white-box testing ketika melakukan pengujian dengan frekuensi yang lebih sedikit.

Dan juga dari hasil data analisis, peneliti menyimpulkan bahwa white-box testing lebih baik dibandingkan manual testing untuk testing dengan frekuensi yang lebih banyak. Dari data juga memperlihatkan kepada kita white-box testing dan manual testing punya kelebihan dan kelemahan masing-masing, bergantung dari kebutuhan mana yang dibutuhkan.

Dalam konteks dimana perusahaan membuat sebuah produk, tentu white-box testing lebih

menguntungkan dikarenakan setiap versi yang dirilis oleh perusahaan tersebut, testing dapat dilakukan secara otomatis tanpa human error. Dimana hal ini sangat efisiensi waktu dan biaya dimana pengujian akan berlangsung sangat singkat.

Namun dalam konteks dimana perusahaan merupakan sebuah vendor pembuat project maka manual testing akan lebih cepat dan menguntungkan dikarenakan perusahaan sebenarnya tidak begitu sering memerlukan pengujian, ketika project yang dibuat dan di-test berhasil sesuai test script yang ada, maka project tersebut bisa dinyatakan berhasil dan siap diluncurkan.

IV. REFERENSI

- [1] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, 2014, doi: 10.1007/s12599-014-0334-4.
- [2] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, 2012, doi: 10.5121/ijesa.2012.2204.
- [3] S. Rizky, "Konsep Dasar Rekayasa Perangkat Lunak," in *Konsep Dasar Rekayasa Perangkat Lunak*, 2011.
- [4] K. Arianto, M. A., Munir, S., dan Khotimah, "ANALISIS DAN PERANCANGAN REPRESENTATIONAL STATE TRANSFER (REST) WEB SERVICE SISTEM INFORMASI AKADEMIK STT TERPADU NURUL FIKRI MENGGUNAKAN YII FRAMEWORK," *J. Teknol. Terpadu*, 2016.
- [5] T. Schweighofer and M. Heričko, "Mobile device and technology characteristics' impact on Mobile application testing," 2013.
- [6] S. Thummalapenta, T. Xie, N. Tillmann, J. De Halleux, and W. Schulte, "MSeqGen: Object-oriented unit-test generation via mining source code," 2009, doi: 10.1145/1595696.1595725.
- [7] N. N. Septiyani and others, "PENGARUH PROGRAM ONE STUDENT ONE PC TERHADAP HASIL BELAJAR PADA MATA PELAJARAN INSTALASI SISTEM OPERASI BERBASIS GUI SISWA KELAS XI JURUSAN TKJ DI

SMK SLAMET RIYADI GEMOLONG
KABUPATEN SRAGEN,” Universitas
Negeri Semarang, 2017.

[8] M. A. Salsburg, “Moving enterprise

software applications to a cloud domain.”
Google Patents, 2014.