



# **BLACKBOX**

# **TESTING**

## **TEORI DAN STUDI KASUS**



**Soetam Rizky Wicaksono (ed.)**

**Kontributor**

**Program Studi Sistem Informasi  
Universitas Ma Chung**

# ***Blackbox Testing***

## ***Teori dan Studi Kasus***

**Program Studi Sistem Informasi  
Fakultas Sains dan Teknologi  
Universitas Ma Chung**

**Soetam Rizky Wicaksono (ed)**



# ***Blackbox Testing***

## ***Teori dan Studi Kasus***

### **Editor / Tata Letak:**

Soetam Rizky Wicaksono

### **Kontributor:**

Program Studi Sistem Informasi  
Fakultas Sains dan Teknologi  
Universitas Ma Chung

### **Penerbit**

CV. Seribu Bintang  
Malang – Jawa Timur - Indonesia  
Website: [www.SeribuBintang.co.id](http://www.SeribuBintang.co.id)  
email : [info@seribubintang.co.id](mailto:info@seribubintang.co.id)  
FB : [www.fb.com/cv.seribu.bintang](https://www.fb.com/cv.seribu.bintang)  
Katalog: <https://seribubintang.gumroad.com>  
Anggota IKAPI no. 320/JTI/2021

**ISBN : 978-623-7000-40-2**

**Edisi Pertama, Juli 2021**

**Hak Cipta dilindungi oleh Undang-undang**

# Pendahuluan

Testing dalam siklus pengembangan perangkat lunak merupakan salah satu unsur yang kerap dilupakan. Bahkan buku referensi khusus untuk testing perangkat lunak berbahasa Indonesia, hingga pada saat buku ini diterbitkan, masih belum ada. Mayoritas materi testing menjadi satu dengan materi rekayasa perangkat lunak atau pengembangan perangkat lunak.

Sehingga pada saat pelaksanaan mata kuliah Testing Perangkat Lunak di program studi Sistem Informasi Universitas Ma Chung, dibuatlah sebuah tugas berseri, dimulai dari Kuis Besar I hingga Kuis Besar IV dan berujung di ujian akhir semester. Tugas tersebut dimulai dari penulisan pendahuluan hingga proses testing sesuai dengan kaidah teoritis terhadap beberapa situs dan aplikasi pembelajaran.

Dikarenakan keterbatasan ruang dan waktu, maka testing yang dilakukan hanya pada proses *blackbox testing* yang diharapkan dapat membuka wawasan bagi para akademisi maupun praktisi dalam proses pengembangan perangkat lunak. Proses testing ini bukan sebagai *judgement* ataupun penilaian, namun lebih kepada

Sebagai bagian dari ujian akhir semester pula, diharapkan bahwa hasil ujian yang umumnya berupa makalah, tidak lagi hanya tergeletak di

## ***Pendahuluan***

---

meja dosen dan menjadi kertas bekas, namun dapat lebih bermanfaat dengan menjadikannya sebagai sebuah buku bunga rampai studi kasus.

Beberapa studi kasus yang cukup menarik untuk ditelaah di dalam buku ini diantaranya: rasionalisasi karyawan di perbankan akibat adanya sistem baru, ataupun sistem rekrutmen online yang bisa berhasil ataupun gagal dalam pelaksanaannya, baik di BUMN maupun di pemerintahan. Semoga studi kasus dalam buku ini dapat bermanfaat bagi keperluan bidang sumber daya manusia di Indonesia.

Selamat berkarya....

Editor,

Dr. Soetam Rizky Wicaksono

# Daftar Isi

<b>Pendahuluan .....</b>	<b>1</b>
<b>Teori Software Testing.....</b>	<b>5</b>
Sejarah .....	6
Definisi .....	12
Perkembangan Teori.....	13
Daftar Pustaka .....	15
<b>Teori White Box Testing.....</b>	<b>18</b>
Definisi .....	23
Perkembangan Teori.....	24
Daftar Pustaka .....	26
<b>Teori Grey Box Testing .....</b>	<b>28</b>
Pendahuluan.....	30
Definisi .....	32
Perkembangan Teori.....	33
Daftar Pustaka .....	35
<b>Teori Blackbox Testing .....</b>	<b>37</b>
Teori.....	39
Definisi .....	48
Perkembangan Teori .....	49
Daftar Pustaka .....	50
<b>Teori Boundary Value Analysis .....</b>	<b>52</b>
Pendahuluan.....	54
Definisi .....	56

Perkembangan Teori.....	57
Daftar Pustaka .....	58
<b>Teori Equivalence Partitioning .....</b>	<b>60</b>
Pendahuluan.....	62
Definisi .....	64
Perkembangan Teori.....	65
Daftar Pustaka .....	67
<b>Studi Kasus Testing Yoopies.....</b>	<b>69</b>
Pendahuluan.....	70
Pembahasan .....	73
Kesimpulan.....	87
Daftar Pustaka .....	89
<b>Studi Kasus Testing Qasir.....</b>	<b>91</b>
Pendahuluan.....	92
Testing .....	94
Kesimpulan dan Saran .....	135
Daftar Pustaka .....	136
<b>Studi Kasus Website Cakap.com .....</b>	<b>137</b>
Latar Belakang .....	138
Pembahasan .....	140
Simpulan Dan Saran .....	155
Daftar Pustaka .....	158

# Teori Software Testing

**Kontributor:**

Stefanus Ditta  
Richi Andrew S.

321810017  
321810015



## **Sejarah**

Seperti yang diketahui bahwa software testing adalah sebuah software yang digunakan untuk menguji sebuah sistem apakah dapat bekerja dengan baik atau tidak. Lalu pada saat ini telah banyak software testing yang dapat digunakan. Namun sejarah penggunaan software testing akan dijelaskan pada bagian berikut :

### **1. Tahun 1930**

Tahun ini mendasari penemuan atau penciptaan komputasi digital. Hal ini bermula saat mesin kalkulator yang diciptakan dapat menyelesaikan soal-soal matematika. Sebelum terjadinya perang dunia ke dua, banyak negara yang mempersiapkan dan meningkatkan kualitas negara mereka khususnya dalam bidang pertahanan. Sehingga pada tahun ini terciptanya analog komputer yang digunakan untuk meluncurkan serangan secara otomatis menggunakan alat tersebut. Lalu semakin dekat dengan perang dunia kedua banyak negara banyak menemukan teknologi yang dapat membantu di dalam perang besar nantinya sehingga pada fase

ini merupakan fase awal munculnya komputer dan software. (Jones, 2013)

## 2. Tahun 1940

Saat perang dunia kedua dimulai, setiap negara mulai menggunakan teknologi yang ditemukan sebelumnya untuk memulai perang dunia kedua. Di mana komputer dan software awal tercipta pada fase yang fungsinya digunakan untuk perang dan tidak selengkap komputer yang kita kenal sekarang. Lalu dalam fase ini juga muncul awal mula ditemukannya software testing. Di aman untuk melancarkan serangan dan juga mencari solusi sebuah komputer dan software pada masa perang dunia ke 2 juga diuji untuk dapat mengetahui apakah terdapat error atau bug yang muncul, karena dapat menghambat jalannya perang. Dalam perang ini, pihak sekutu yang tidak ingin dirinya kalah berusaha keras untuk menemukan sebuah teknologi yang dapat digunakan untuk membaca kode militer tentara Jepang dan Jerman dan akhirnya Jerman dan Jepang mengakui kekalahan. Lalu pada fase ini pula kata "*computer*" melekat pada orang yang mahir dalam bidang matematika, namun menurut

perkembangan jaman dan teknologi yang pesat setelah perang dunia kedua, istilah komputer menjadi istilah yang kita kenal sekarang

### 3. Tahun 1950

Pada saat itu telah banyak perusahaan yang bergerak di bagian komputer yang menggunakannya. Penggunaannya pada saat itu bertujuan untuk mengurangi atau bahkan menghilangkan error pada sebuah sistem. Agar sistem tersebut dapat berjalan dengan baik.

### 4. Tahun 1960

Pada tahun ini penggunaan software testing semakin meningkat dan juga diikuti oleh banyaknya variasi program yang ada. Sehingga yang sebelumnya software testing dapat bekerja dengan mudah karena sistem yang di tes tidak besar, pada tahun ini banyak sistem yang bermunculan dengan bobot yang lebih besar sehingga membuat software testing berjuang untuk berkembang menurut perkembangan sistem dan aplikasi yang muncul.

## 5. Tahun 1970

Pada tahun 1970 *Waterfall Models* di dokumentasikan pertama kali oleh Royce sebagai *Life Cycle Model* pertama yang dapat di dokumentasikan ke publik. *Waterfall Models* sendiri merupakan versi populer dari siklus hidup pengembangan sistem untuk rekayasa perangkat lunak. Dan di dalam *Waterfall Models* sendiri seluruh pengembangan *Software* dibagi menjadi fase proses yang terpisah.

## 6. Tahun 1980

Pada tahun 1980 pengujian diintegrasikan dengan sebuah konsep baru yaitu *Defect Prevention*. Untuk melakukan sebuah pengujian, diperlukan metodologi yang sesuai, dan pengujian tersebut harus dilakukan di seluruh *Software Development Life Cycle* dengan manajemen yang tepat. Dan pada tahun 1980-an di temukannya sebuah model yang memiliki peranan penting dalam proses pengujian perangkat lunak. Yaitu V-Model atau dikenal sebagai Model Verifikasi dan Validasi. Dan diperkenalkan oleh Paul Rook. V-Model adalah proses (Singh, 2013)

7. Tahun 1986

Pada tahun 1986 *Spirall Models* pertama kali dijelaskan oleh Barry Boehm dalam makalahnya pada tahun 1986 "Model Spiral Pengembangan dan Peningkatan Perangkat Lunak". Namun pada tahun 1988 Boehm menerbitkan makalah yang serupa. Model ini bukan merupakan model yang pertama namun model pertama yang menjelaskan mengapa iterasi penting. (Essays, 2016)

8. Tahun 1990

Pada tahun 1990 , alat testing yang lebih canggih sudah lebih kaya akan bahasa skrip. Alat pengetesan membantu dalam mengelola semua artefak dari tahap persyaratan dan menguji semua jenis skrip dan kesalahan. Dan pada pertengahan 1990-an, dengan semakin populernya internet, software dikembangkan tanpa standar testing software, menjadikannya lebih kompleks untuk pengujian. Seperti halnya dokumen tidak dapat dijual tanpa mendesain setiap hasil akhir dari setiap langkah peninjauan, dan dengan cara yang sama testing tidak dapat dilakukan. Pendekatan

baru yang didefinisikan untuk menguji skenario ini disebut Pendekatan *Agile Testing*.

## **Definisi**

Pengertian dari software testing berdasarkan teori ANSI/IEEE 1059 adalah proses menganalisis suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (defects / error / bos) dan mengevaluasi fitur-fitur dari entitas software. (Singh and Kumar Singh, 2019)

Menurut Myers pada tahun 1972 software testing adalah sebuah proses eksekusi program atau sistem secara intens untuk menemukan suatu error. (David Craig and P. Jaskiel, 2002)

Kemudian dalam sisi software engineering berdasarkan IEEE adalah suatu penerapan pendekatan yang secara sistematis, disiplin, dan terukur untuk proses dari pengembangan, operasi, dan pemeliharaan suatu sistem perangkat lunak. (Atherstone, 2003)

Sedangkan menurut Fritz Bauer software Engineering adalah pembentukan dan penggunaan prinsip rekayasa (engineering) untuk mendapatkan perangkat lunak secara ekonomis namun andal dan dapat bekerja secara efisien pada komputer. (S. Pressman, 2005)

## Perkembangan Teori

Jurnal "Aplikasi Metode-Metode Software Testing pada Configuration, Compatibility dan Usability Perangkat Lunak " yang ditulis oleh Suhatati Tjandra dan C. Pickerling pada tahun 2015 tentang software testing itu merupakan komponen dari Software Quality Control (SQC) dan Software Quality Assurance, jadi kualitas dari suatu software ditentukan dari produknya dan juga dari prosesnya. Jadi diharapkan dengan software testing, software yang dikembangkan dapat menjadi software yang berkualitas, dan sesuai dengan kebutuhan dari stakeholder, usability nya tinggi, dan dapat meningkatkan kinerja dari user. (Tjandra and Pickerling, 2015)

Jurnal "Pengujian Perangkat Lunak" yang disusun oleh Linda Liana mengatakan bahwa pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas dari perangkat lunak tidak dapat terlalu ditekan karena manusia tidak bisa lepas dari kesalahan. Oleh karena itu diperlukan jaminan kualitas dengan adanya pengujian itu sendiri. Meningkatnya visibilitas perangkat lunak sebagai suatu elemen

sistem dan adanya biaya yang muncul dikarenakan munculnya dari kegagalan suatu software. Hal tersebut memicu munculnya perencanaan yang baik melalui software testing yang teliti. (Liana, 2015)

## Daftar Pustaka

- Atherstone, A. (2003) 'The Martyrs' Memorial at Oxford', *Journal of Ecclesiastical History*, 54(2), pp. 278–301. doi: 10.1017/S0022046902005638.
- David Craig, R. and P. Jaskiel, S. (2002) *Systematic Software Testing - Rick David Craig, Stefan P. Jaskiel - Google Buku*. United States of America. Available at: [https://books.google.co.id/books?id=2\\_gbZYZcZXgC&pg=PA3&dq=definition+software+testing+myers&hl=id&sa=X&ved=0ahUKewjvq7OYxvzoAhUSSX0KHROFAPiQ6AEIKDAA#v=onepage&q=definition+software+testing+myers&f=false](https://books.google.co.id/books?id=2_gbZYZcZXgC&pg=PA3&dq=definition+software+testing+myers&hl=id&sa=X&ved=0ahUKewjvq7OYxvzoAhUSSX0KHROFAPiQ6AEIKDAA#v=onepage&q=definition+software+testing+myers&f=false) (Accessed: 23 April 2020).
- Essays, U. (2016) *Spiral Model | History | The Spiral Model | Applications*. Available at: <https://www.ukessays.com/essays/information-technology/the-spiral-life-cycle-model-information-technology-essay.php> (Accessed: 13 March 2020).
- Jones, C. (2013) *The Technical and Social History of Software Engineering: Proquest Tech & Business Books*. Available at: [https://books.google.co.id/books?id=\\_H8lAgAAQBAJ&pg=PA37&lpg=PA37&dq=history+software+engineering+%221930%22&source=bl&ots=q1v64rCzIz&sig=ACfU3U2g1uIHMHHL8P8fcnxjBE0cUz5Cvg&hl=id&sa=X&ved=2ahUKewj0t47EgfvoAhVEg-YKHa1xBKoQ6AEwFnoECACQAQ#v=onepage&q=history](https://books.google.co.id/books?id=_H8lAgAAQBAJ&pg=PA37&lpg=PA37&dq=history+software+engineering+%221930%22&source=bl&ots=q1v64rCzIz&sig=ACfU3U2g1uIHMHHL8P8fcnxjBE0cUz5Cvg&hl=id&sa=X&ved=2ahUKewj0t47EgfvoAhVEg-YKHa1xBKoQ6AEwFnoECACQAQ#v=onepage&q=history) (Accessed: 23 April 2020).

- Liana, L. (2015) 'Software testing', *Software Testing*, 9781107012, pp. 1–626. doi: 10.1017/CBO9781139196185.
- S. Pressman, R. (2005) *Software Engineering: A Practitioner's Approach - Roger S. Pressman - Google Buku*. Available at: [https://books.google.co.id/books?id=bL7QZHTwvaUC&pg=PA21&dq=definition+software+engineering+fritz+bauer&hl=id&sa=X&ved=0ahUKEwikoKn0\\_zoAhXZXisKHeKjC4Q6AEINDAB#v=onepage&q=definition software engineering fritz bauer&f=false](https://books.google.co.id/books?id=bL7QZHTwvaUC&pg=PA21&dq=definition+software+engineering+fritz+bauer&hl=id&sa=X&ved=0ahUKEwikoKn0_zoAhXZXisKHeKjC4Q6AEINDAB#v=onepage&q=definition software engineering fritz bauer&f=false) (Accessed: 23 April 2020).
- Singh, A. and Kumar Singh, S. (2019) *SOFTWARE TESTING - Dr. Sanjay Kumar Singh and Dr. Amarjeet Singh - Google Buku*. Available at: [https://books.google.co.id/books?id=HdKwDwAAQBAJ&pg=PT6&dq=definition+software+testing+ANSI/IEEE+1059&hl=id&sa=X&ved=0ahUKEwjQrs7Ot\\_zoAhUVeysKHR9QA3YQ6AEIMjAB#v=onepage&q&f=false](https://books.google.co.id/books?id=HdKwDwAAQBAJ&pg=PT6&dq=definition+software+testing+ANSI/IEEE+1059&hl=id&sa=X&ved=0ahUKEwjQrs7Ot_zoAhUVeysKHR9QA3YQ6AEIMjAB#v=onepage&q&f=false) (Accessed: 22 April 2020).
- Singh, M. (2013) *Perbedaan antara model-V dan model Air Terjun - GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/difference-between-v-model-and-waterfall-model/> (Accessed: 13 March 2020).
- Tjandra, S. and Pickerling, C. (2015) 'Aplikasi Metode-Metode Software Testing Pada Configuration , Compatibility Dan Usability Perangkat Lunak', *IDeaTech*, pp. 367–374.



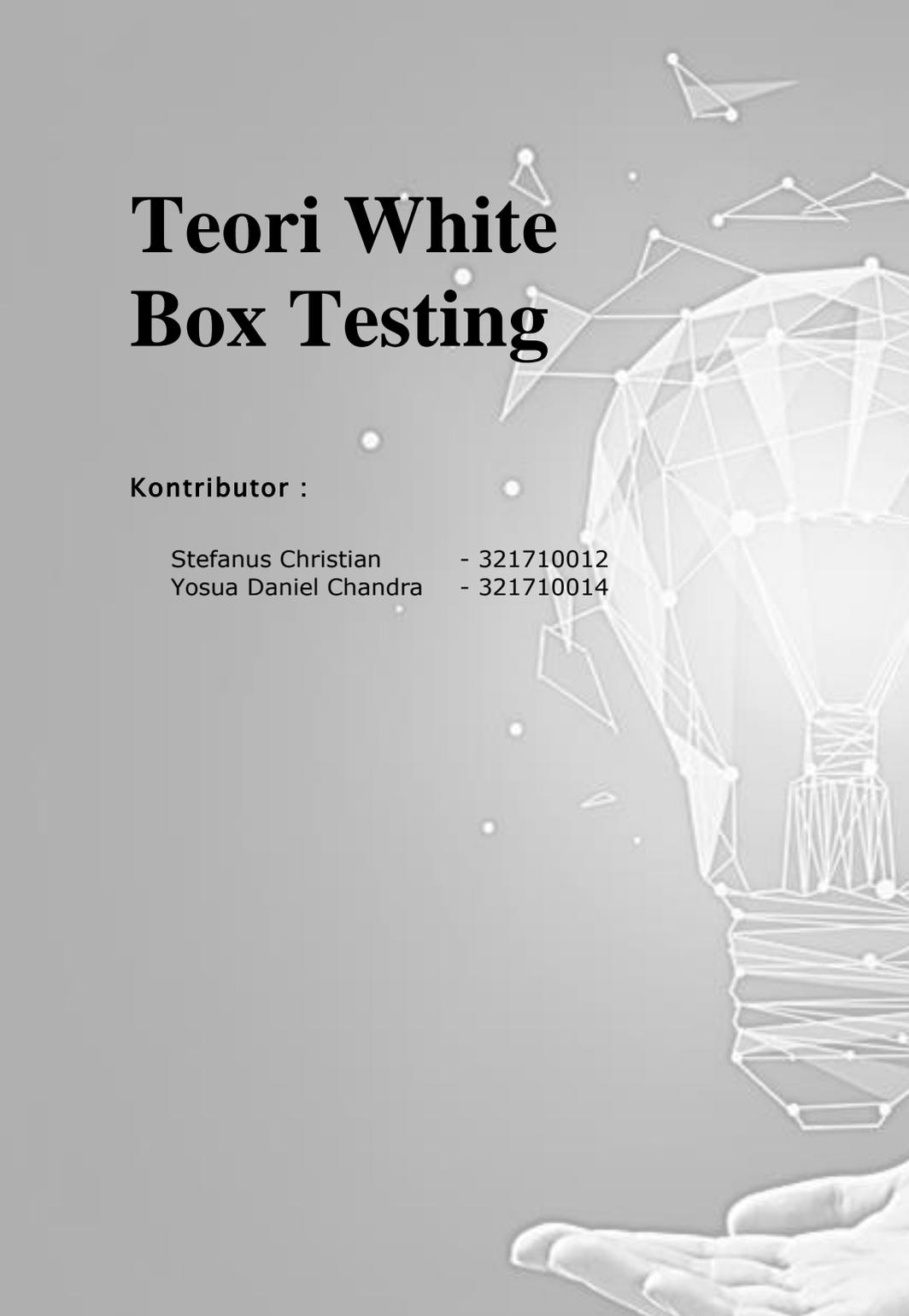
---

Kemarin aku pandai, sehingga aku ingin mengubah dunia. Hari ini aku bijaksana, sehingga aku ingin mengubah diriku sendiri.

---

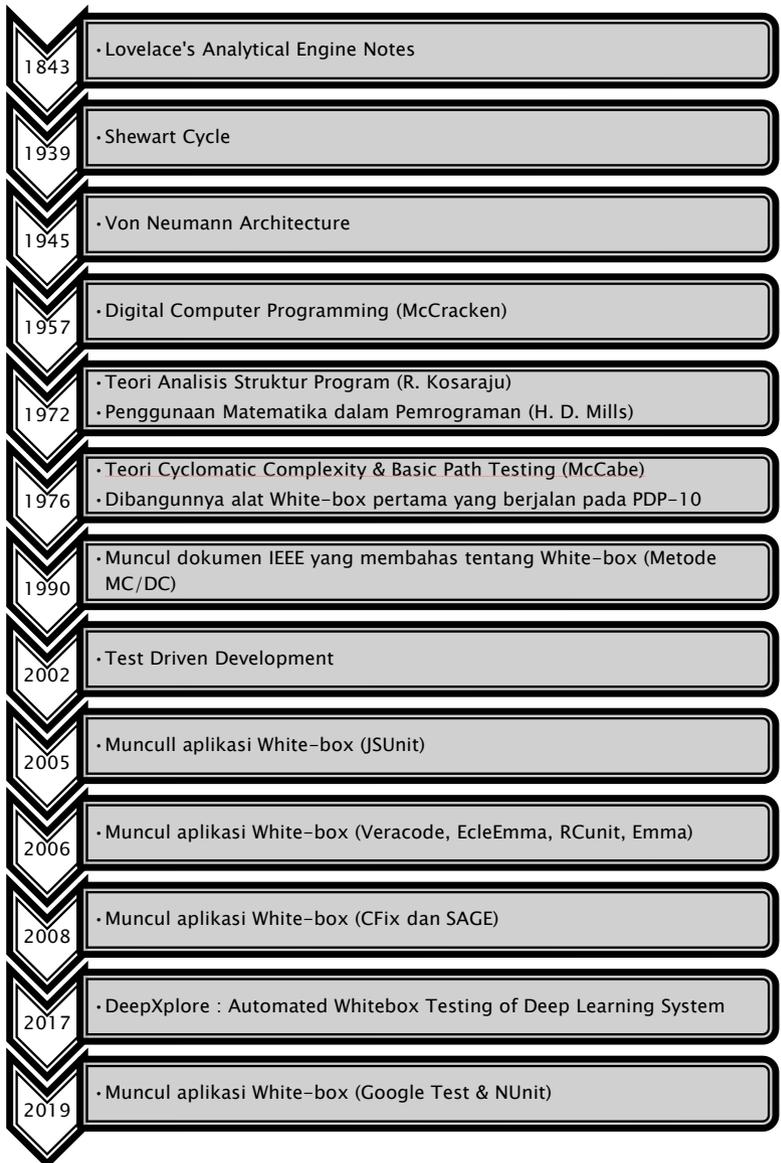


# Teori White Box Testing



**Kontributor :**

Stefanus Christian - 321710012  
Yosua Daniel Chandra - 321710014



Pengujian *White-box* adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja suatu aplikasi, sejarah pengujian *white-box* dimulai dari tahun 1843 Lovelace menambahkan satu set catatan. Catatan mencakup metode untuk menghitung urutan nomor Bernoulli dengan Mesin. Berdasarkan karya ini, Lovelace sekarang secara luas diakui sebagai programmer komputer pertama dan metodenya diakui sebagai program komputer pertama di dunia. 1939 Walter Andrew Shewhart membahas tentang *Scientific Method of Improvement* di buku keduanya tentang Metode Statistik dari Sudut Pandang Kendali Mutu. Shewart Cycle inilah yang kemudian dipopulerkan menjadi metode Plan Do Check Act, yang membantu melakukan pengujian secara terencana dan terstruktur. 1945 Dalam makalah First Draft of Report on EDVAC, John von Neumann menerbitkan deskripsi desain logis komputer menggunakan konsep program stored-procedure. Ini adalah desain yang menjadi dasar kebanyakan komputer saat ini dimana setiap *command* dijalankan melalui beberapa prosedur. 1957 didalam buku Pemrograman Komputer Digital, McCracken membahas sejumlah teknik

pemrograman seperti checkout program dan penggunaan *break point* dalam suatu program di mana fungsi program dapat diverifikasi.(Meerts, 2020)

1972 dengan munculnya teori tentang analisis struktur program yang ditulis oleh R. Kosaraju dan teori tentang penggunaan matematika dalam stuktur pemrograman. Dua teori ini akhirnya dipakai oleh Thomas J. McCabe dalam mengembangkan teorinya yang berjudul *Cyclomatic Complexity*. Penelitian pertama McCabe ini digunakan untuk membangun sebuah alat yang bekerja pada PDP-10 untuk menganalisis struktur program Fortran. PDP-10 yang dipakai merupakan kerangka utama computer yang diproduksi mulai dari tahun 1966 dan akhirnya berhenti produksi pada tahun 1983 (*Definitions.net*. STANDS4 LLC, 2020). Pada tahun 1990, IEEE memunculkan dokumen yang berkaitan dengan *White-box testing* dengan menggunakan metode *Modified Condition/Decision Coverage* (MC/DC).

Pada tahun 2000 sampai 2019 muncul beberapa aplikasi yang digunakan untuk melakukan testing secara otomatis. Contohnya di tahun 2000 muncul aplikasi CppUnit dan Junit,

kemudian lima tahun kemudian muncul JUnit. Pada tahun berikutnya dirilis 4 aplikasi testing lainnya seperti Veracode, EcleEmma, RCunit, dan Emma. Di tahun 2008 juga ada aplikasi CFix dan SAGE yang dipublikasikan, tetapi untuk SAGE ini memiliki keunikan karena perangkat lunak ini lebih fokus dalam memeriksa logika keamanan dari sebuah aplikasi. SAGE ini didirikan oleh peneliti dari Microsoft yang berkerjasama dengan UC Berkley. (Godefroid, 2009)

Di tahun 2017, munculah DeepXplore yang merupakan kerjasama dari Universitas Collumbia dan Universitas Lehigh. Perangkat lunak ini merupakan alat tes yang menggunakan kerangka kerja *White-box* dan sudah terotomatisasi dengan menggunakan sistem *Deep Learning*. Sistem ini digunakan dalam membuat fitur *auto-pilot* di mobil-mobil canggih serta dalam mendeteksi virus. Kemudian tahun 2019 munculah aplikasi besutan dari google yang bernama Google Test dan juga aplikasi NUnit. (Pei, Cao, Yang, & Jana, 2019)

## Definisi

Menurut Srinivasan Desikan dengan Gopaldaswamy Ramesh dalam bukunya yang berjudul "Software Testing: Principles and Practice" mendefinisikan *white box testing* sebagai cara melakukan pengujian fungsi eksternal dari bahasa pemrograman dengan memahami dan melakukan tes struktur koding sebuah program yang mencerminkan fungsi eksternal.

Sedangkan menurut sumber lain yang bersasal dari sebuah jurnal "Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing" yang ditulis oleh Yudie Irawan menyatakan bahwa *white box* adalah metode desain *test case* yang menggunakan struktur kendali dari desain prosedural.

*White box testing* juga didefinisikan sebagai pengujian struktural yang berarti menguji sebuah perangkat lunak berdasarkan struktur logika yang dipakai dalam sistem tersebut. (Stein, R. Timothy, 2006).

## Perkembangan Teori

Penerapan teori *white-box testing* semakin meningkat di tahun 2017. Sebuah jurnal "DeepXplore: Automated Whitebox Testing of Deep Learning System" yang ditulis oleh Kexin Pei, Yinzhi Cao, Junfeng Yao, dan Suman Jana yang merupakan mahasiswa gabungan dari Universitas Columbia dan Universitas Lehigh membahas tentang pengembangan sebuah sistem Deep Learning (DL) yang merupakan perkembangan dari framework whitebox. Sistem ini mengembangkan model algoritma whitebox menjadi algoritma DNN, yaitu algoritma yang memiliki struktur seperti neuron atau sel otak manusia. Pengembangan sistem ini dibuat pada tahun 2017 guna untuk melakukan testing terhadap mode auto-pilot pada mobil Tesla. (Pei et al., 2019)

Penelitian selanjutnya yaitu dalam Jurnal berjudul "Testing Database Applications with SQL Semantics" yang dilakukan oleh M.Y. Chan and S.C. Cheung dari Universitas Sains dan Teknologi Hongkong membahas tentang bagaimana melakukan testing terhadap aplikasi database yang

menggunakan. Penelitian ini dilakukan karena logika SQL dalam aplikasi database seringkali terlewat oleh algoritma whitebox. Maka dari itu dalam jurnal ini akan dilakukan pendekatan terhadap logika SQL agar bisa diubah dalam bahasa yang umum agar bisa ditangkap dan dimengerti oleh algoritma whitebox. (Chan & Cheung, 2019)

Kedua penelitian tersebut memanfaatkan Teknik *white-box testing* yang dimodifikasi dengan cabang keilmuan lain seperti *deep learning* dan metode *WHODATE* untuk meningkatkan keefektifan pengujian aplikasi terkait, sehingga luaran dan tujuan pengujian aplikasi tersebut lebih sesuai sasaran dan presisi.

## Daftar Pustaka

- Chan, M. Y., & Cheung, S. C. (2019). Testing database applications with SQL semantics. *Proceedings of 2nd International Symposium on Cooperative Database Systems for Advanced Applications*, (March), 363–374.
- Godefroid, P. (2009). *Software Model Checking Improving Security of a Billion Computers*. (July), 1–1. [https://doi.org/10.1007/978-3-642-02652-2\\_1](https://doi.org/10.1007/978-3-642-02652-2_1)
- Journal Speed - Sentra Penelitian Engineering dan Edukasi - Volume 9 No 3, Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing (2017), Yudie Irawan (Diakses: 5 Maret 2020)
- Kompasiana (2010), White Box Testing, tersedia pada: [https://www.kompasiana.com/diskbrain\\_bih/55004dc4813311791bfa74d0/white-box-testing](https://www.kompasiana.com/diskbrain_bih/55004dc4813311791bfa74d0/white-box-testing) (Diakses: 5 Maret 2020)
- Meerts, J. (2020). Testing References - The History of Software Testing. Retrieved April 28, 2020, from 2019 website: <http://www.testingreferences.com/testinghistory.php>
- Pei, K., Cao, Y., Yang, J., & Jana, S. (2019). Deepxplore: Automated whitebox testing of deep learning systems. *Communications of the ACM*, 62(11), 137–145. <https://doi.org/10.1145/3361566>
- Repository Universitas Ahmad Dahlan (2020), White Box Testing, tersedia pada:

<http://eprints.uad.ac.id/7752/1/5-white-box.pdf> (Diakses: 5 Maret 2020).

Software Testing: Principles and Practice (2003), Srinivasan Desikan, Gopaldaswamy Ramesh (Diakses: 5 Maret 2020)

The Computer System Risk Management and Validation Life Cycle (2006), R. Timothy Stein, (Diakses: 5 maret 2020)

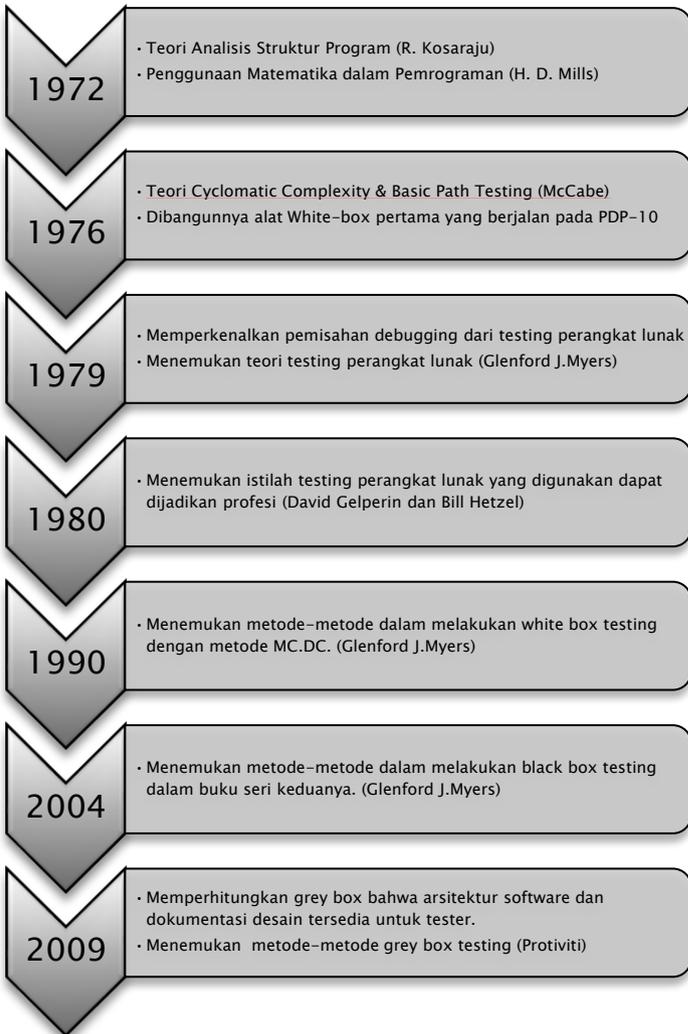
# Teori Grey Box Testing

**Kontributor:**

Shalsha Abhiel – 321710011

Yuri Amelia - 321710015





## **Pendahuluan**

Tahun 1972 ditemukan teori tentang analisis struktur program yang ditulis oleh R. Kosaraju dan ditemukan teori tentang penggunaan matematika dalam struktur pemrograman. Yang dipakai oleh Thomas J. McCabe pada tahun 1976 dalam mengembangkan teorinya yang berjudul Cyclomatic Complexity dan melakukan penelitian pertama McCabe ini membangun sebuah alat yang bekerja pada PDP-10 untuk menganalisis struktur program Fortran.

Kemudian pada tahun 1979 dari buku *The Art of Software Testing*, oleh Glenford J. Myers memperkenalkan pemisahan antara debugging dari testing untuk mendeteksi kesalahan dengan lebih mudah pada perangkat lunak. Sehingga beliau dapat menemukan teori-teori testing perangkat lunak. (Myers, 1979)

Pada tahun 1980, David Gelperin dan Bill Hetzel menemukan istilah testing perangkat lunak yang digunakan secara umum dan berpendapat jika testing perangkat lunak dapat dijadikan sebagai profesi. (Gelperin & Hetzel, 1980)

Pada tahun 1990, Glenford J. Myers memunculkan dokumen IEEE yang berkaitan dengan *White-box testing* dengan menggunakan metode *Modified Condition/Decision Coverage* (MC/DC). Kemudian ditahun 2004 Badgett, Todd Thomas, dan Corey Sandler melakukan testing perangkat lunak dengan menggunakan metode equivalence partitioning. Yaitu; Graph test, equivalence partitioning, serta boundary value analysis Pada buku serinya yang kedua. (Badgett et al., 2004)

Pada tahun 2009, Saleh memperhitungkan grey box bahwa arsitektur software dan dokumentasi desain tersedia untuk tester. Serta menemukan metode-metode grey box testing mengenai fase-fase pengujian. (Saleh,2009)

## Definisi

*Software testing* adalah proses menemukan bukti dari kesalahan – kesalahan yang terdapat di dalam software dengan memberikan statement bahwa beberapa orang mengetes software juga bertanggungjawab untuk melakukan *debug* terhadap software tersebut. (McGregor & Sykes, 2001)

*Grey Box Testing* adalah sebuah metodologi kombinasi dari *Black Box* dan *White Box Testing*, menguji *software* berdasarkan spesifikasi tetapi menggunakan cara kerja dari dalam. *Grey Box* dapat di gunakan dengan baik dalam *software testing*. (Rosi, 2016)

*Grey box testing* diasumsikan bahwa arsitektur *software* dan dokumentasi desain tersedia untuk penguji. *Grey Box* juga berdasarkan desain dokumen dan *functional specifications*. (Saleh, 2009)

Dari definisi tersebut dapat disimpulkan bahwa *Grey Box Testing* adalah teknik pengujian system dengan pengetahuan yang terbatas dari internal system yang memiliki akses ke dokumen desain rinci.

## Perkembangan Teori

Berdasarkan penelitian pada jurnal yang berjudul "Aplikasi Metode-Metode Software Testing pada Configuration, Compatibility, dan Usability Perangkat Lunak" yang ditulis oleh Suhatati Tjandra dan C. Pickerling, Teknik Informatika, Sekolah Tinggi Teknik Surabaya. Berisi mengenai testing dengan menggunakan grey box yang menyatukan antara black-box testing dan white-box testing untuk menguji halaman web. Dalam web testing ini yang dilakukan adalah program dalam database skala besar atau internet web pages. Testing yang dilakukan yaitu menggunakan compatibility, usability dan grey-box testing. Bahwa grey box testing digunakan oleh seorang testier menguji software untuk melakukan pengujian terhadap web pages dan juga melibatkan source code untuk mengetahui apa yang membuat software dapat bekerja atau tidak karena pada web pages HTML bukan bahasa pemrograman, tetapi markup language. Dengan melakukan grey box testing ini maka dapat dilakukan pengecekan terhadap bug dalam HTML. (Suhatati T & C. Pickerling, 2015)

Berikutnya penelitian pada jurnal yang berjudul "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques" yang ditulis oleh Mohd. Ehmer Khan dan Farmeena Khan. Berisi mengenai testing dengan menggunakan grey-box, tester harus memiliki pengetahuan tentang struktur dan alogaritma data internal dalam merancang pengujian tersebut. Dengan teknik pengujian pada grey-box dalam metode white-box dan black-box menggunakan model arsitektur, bahasa permodelan terpadu, model negara. (Mohd. Ehmer Khan & Farmeena Khan, 2012)

Kedua penelitian tersebut memanfaatkan teknik grey-box testing pada software yang dapat memberikan manfaat gabungan dari teknik white-box dan teknik black-box. Dalam pengujian grey-box testing tersebut, tester dapat merancang skenario pengujian yang sangat baik, dengan sudut pandang pengguna dan bukan dari sudut pandang perancang.

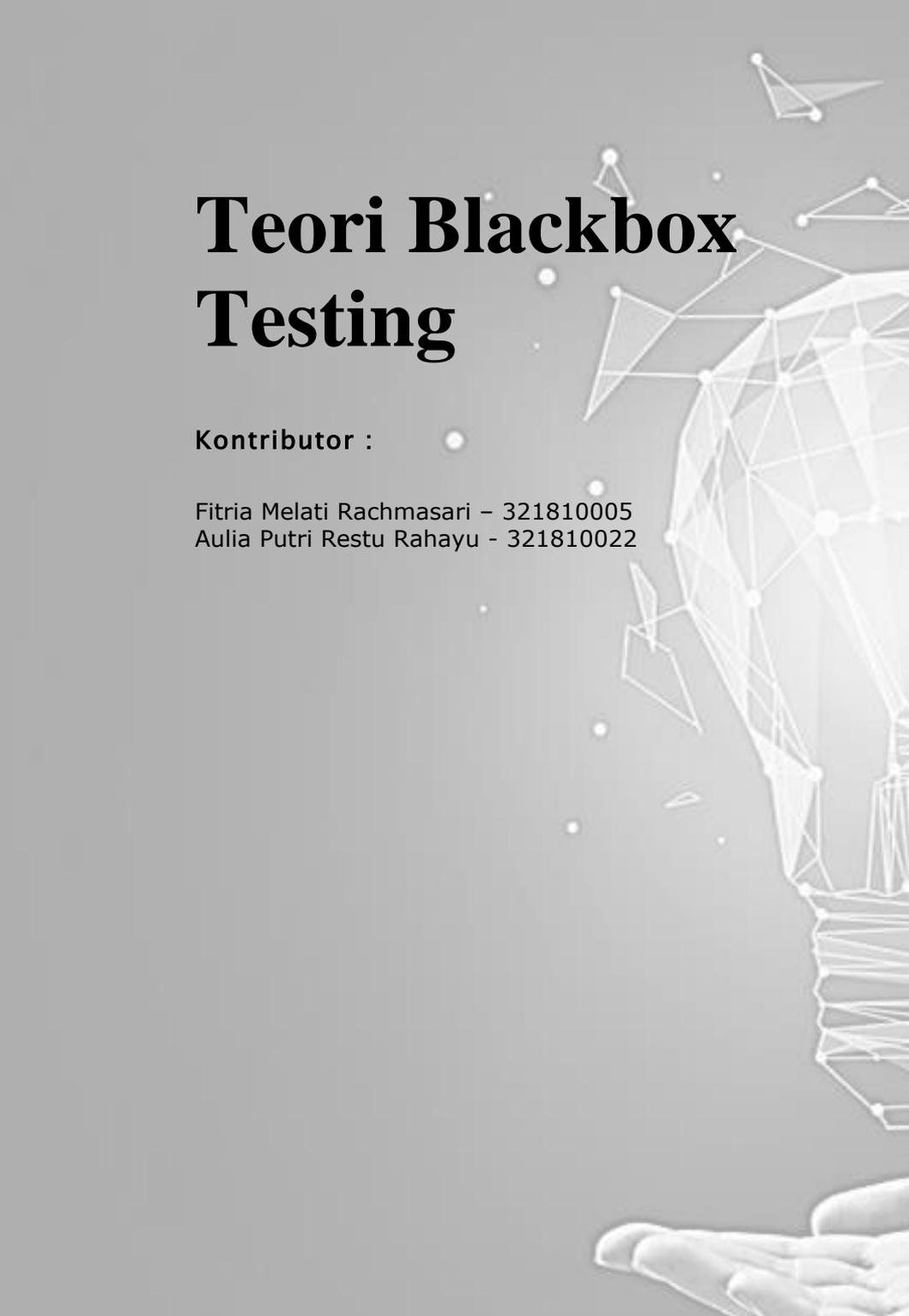
## Daftar Pustaka

- Myers, G. J. (1979). The Art Of Software Testing. In *IEEE Proceedings of the National Aerospace and Electronics Conference* (First Edit, Vol. 2). John Wiley & Sons, Inc. <https://doi.org/10.1109/naecon.1991.165837>
- Gelperin, D., & Hetzel, B. (1980). The Growth Of Software Testing. *Communication Of The ACM*, 31(6), 10. <https://doi.org/10.1145/62959.62965>
- Badgett, Thomas, T. M., & Sandler, C. (2004). The Art Of Software Testing. In John Wiley & Sons, Inc (Second). John Wiley & Sons, Inc. [www.ccsenet.org/cis](http://www.ccsenet.org/cis)
- Saleh. (2009). Software Engineering. Kassem A. Saleh. Grey box testing, 224-225. <https://books.google.co.id/books?id=N69KpjBEWygC&pg=PA224&dq=grey+box+testing+saleh&hl=en&sa=X&ved=0ahUKEwj55-Kwh5zpAhVBT30KHbsTBVEQ6AEIKTAA#v=onepage&q=grey%20box%20testing%20saleh&f=false>
- Rosi, E. (2016). STRATEGI, TEKNIK, FAKTOR PENDUKUNG DAN PENGHAMBAT PENGUJIAN UNTUK PENGEMBANG PERANGKAT LUNAK PEMULA. *STRATEGI, TEKNIK, FAKTOR PENDUKUNG DAN PENGHAMBAT PENGUJIAN UNTUK PENGEMBANG PERANGKAT LUNAK PEMULA*.
- Suhatati T & C. Pickerling, (2015). Aplikasi Metode-metode software testing pada configuration, compatibility, dan usability perangkat lunak. <https://www.researchgate.net/publication/27>

0554162\_A\_Comparative\_Study\_of\_White\_Box\_Black\_Box\_and\_Grey\_Box\_Testing\_Techniques

Mohd. Ehmer Khan & Farmeena Khan, (2012). A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. [https://www.researchgate.net/publication/270554162\\_A\\_Comparative\\_Study\\_of\\_White\\_Box\\_Black\\_Box\\_and\\_Grey\\_Box\\_Testing\\_Techniques](https://www.researchgate.net/publication/270554162_A_Comparative_Study_of_White_Box_Black_Box_and_Grey_Box_Testing_Techniques)

# Teori Blackbox Testing

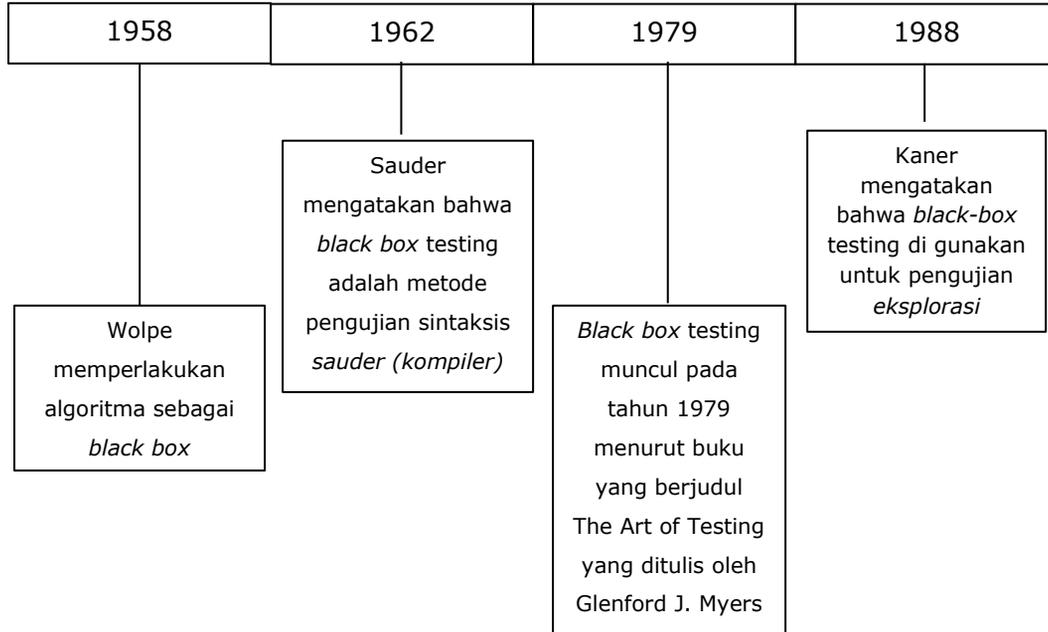


**Kontributor :**

Fitria Melati Rachmasari – 321810005  
Aulia Putri Restu Rahayu - 321810022

## Teori Blackbox Testing

---



## Teori

Pada tahun 1958 Wolpe memperlakukan algoritma sebagai *black box*. Lalu pada tahun 1962 Sauder mengatakan bahwa *black box* testing adalah metode pengujian sintaksis sauder (kompiler). (Murnane, 2009)

Menurut buku yang berjudul *The Art of Testing* yang ditulis oleh Glenford J. Myers (2004) menyebutkan bahwa *black box* testing muncul pada tahun 1979. Di dalam buku tersebut menyebutkan bahwa *black box* testing tidak berkonsentrasi pada perilaku internal dan struktur program tetapi *black box* testing adalah salah satu strategi pengujian penting dalam *software* testing yang berkonsentrasi menemukan keadaan bagaimana program tidak berjalan sesuai dengan spesifikasinya. (Myers et al., 2004)

Selanjutnya pada tahun 1988 Kaner mengatakan bahwa *black box* testing di gunakan untuk pengujian eksplorasi. Berikut metode yang di gunakan untuk melakukan pengujian *black box*, yaitu:

1. *Equivalence Partitioning*
2. *Boundary Value Analysis*
3. *Syntax Testing*

*Syntax Testing* adalah metode untuk memperoleh kasus pengujian dari bidang *input* yang di tentukan secara formal, sering kali dalam sebuah meta bahasa seperti BNF. Spesifikasi dari formal yaitu Abstrak *Syntax Tree* (AST) mengungkapkan hubungan antara *hirarki* utama (non-terminal) dan *hirarki* cabang (terminal). Tes uji yang valid di rancang secara sistematis 'menutupi' cabang-cabang AST, sementara tes yang tidak valid dapat di rancang dengan memasukkan kesalahan ke terminal dan *node nonterminal* AST.

4. *Random Testing*

*Random Testing* adalah metode *black box* testing yang nilai data uji dipilih secara acak dari *input* domain melalui program yang

sedang di uji. Pola pengujian nilai data yang dipilih oleh penguji jika melakukan metode ini secara manual, otomatis dipandang sebagai suatu keharusan. Pada tahun 1940, *pseudo-random number* generator pertama kali di kembangkan.

*Random Testing* merupakan bagian integral dari "*Cleanroom*" IBM dengan metodologi pengembangan perangkat lunak dan juga dianggap sebagai teknik "*crash-proofing*" pada suatu sistem yang mungkin tidak dapat ditemukan di dalam metode pengujian *black box* lainnya. 4 pendekatan utama dari RT ini adalah generasi yang sepenuhnya acak, generasi acak dalam kesetaraan kelas, pengujian kombinasi acak, dan pengujian acak statistik.

5. *Non-prescriptive approaches to black-box testing*

*Non-prescriptive approaches to black-box testing* adalah pendekatan yang tidak terstruktur dan biasanya tanpa ada naskah pengujian yang didasarkan pada

pengetahuan domain yang unik dan pengalaman dari masing-masing penguji.

a. *Error Guessing*

*Error Guessing* diterbitkan oleh Myers pada 1979, pada pengujian ini penguji mengidentifikasi daftar situasi rawan kesalahan dan berasal dari uji kasus yang dapat mendeteksi setiap potensi terjadinya kesalahan.

b. *Exploratory Testing*

*Exploratory Testing* pertama kali ditemukan oleh Kaner pada tahun 1988 untuk menggambarkan proses pengujian terhadap perangkat lunak yang berpengalaman ketika merancang dan melakukan uji kasus non-preskriptif terhadap program. ET melakukan pendekatan pengujian non-preskriptif yang dapat menghasilkan *black* atau *grey*. Di ET setiap *test case* baru dapat di rancang dengan cepat berdasarkan pengetahuan yang diperoleh selama pelaksanaan kasus tes sebelumnya.

6. *Test Catalogues, Test Categories and Test Matrices*

*Test Catalogues, Test Categories and Test Matrices* adalah tiga metode yang dapat digunakan untuk memetakan aturan desain kotak uji untuk memasukkan tipe bidang, merencanakan dan melacak pengujian *black box* testing.

*Test Catalogues, Test Categories and Test Matrices* merupakan solusi sederhana untuk mendokumentasikan konteks di mana aturan desain kasus uji tertentu berlaku dan efektif, memungkinkan informasi ini disimpan dan digunakan kembali pada sistem yang sama atau serupa dan untuk dibagikan dengan penguji lainnya dan juga memungkinkan pengujian menyesuaikan dengan kebutuhan pengujian khusus pada masing-masing program yang sedang diuji. Aturan desain kasus uji yang termasuk dalam katalog, kategori dan matriks dapat dipilih dari metode *black box* testing preskriptif seperti EP, BVA dan ST, pengetahuan penguji berpengalaman dan

dari informasi *grey box* (misalnya pengetahuan kode sumber program).

### ***7. Combinatorial Test Methods***

*Combinatorial Test Methods* memungkinkan pembuatan otomatis *black box* testing melalui penerapan algoritma kombinatorial (yaitu TCCR) untuk menguji nilai data yang ada diturunkan selama penerapan metode seperti EP, BVA, dan ST. Metode pengujian kombinatorial dapat digunakan untuk mengurangi jumlah kasus uji yang dihasilkan oleh metode *black box* testing lainnya.

#### ***a. All Combinations***

*All Combinations* adalah kasus uji yang dipilih dengan mengambil produk *Cartesian n-ary* dari kumpulan data uji serta menghasilkan satu set *tupel* yang di pesan. *All Combinations* juga dikenal sebagai produk langsung dari set dan produk silang yang menghasilkan set *test* yang mencapai cakupan *n-bijaksana*, di mana semua kemungkinan

kombinasi nilai data dari  $n$  bidang dicakup oleh setidaknya satu uji kasus.

*b. Each Choice*

*Each Choice* digunakan untuk memastikan bahwa semua nilai data uji dari set dimasukkan, setidaknya satu tes kasus yang mencapai cakupan *l-wise coverage*. EC telah digunakan di dalam EP, BVA dan pendekatan *Base Choice*.

*c. Base Choice*

*Base Choice* digunakan sebagai pendekatan untuk memilih uji kasus terhadap *Category Partition Method*.

*d. Orthogonal Array Testing*

*Orthogonal Arrays* (OA) berasal dari konsep matematika Kotak Latin. OA adalah *array* dua dimensi di mana ada dua kolom berisi semua kombinasi pasangan nilai dan jika ada pasangan terjadi beberapa kali maka masing-

masing akan muncul persis jumlah yang sama kali.

e. *Specification-Based Mutation Testing*

*Specification-Based Mutation Testing* adalah artefak yang diuji dengan menerapkan sebuah 'operator mutasi' yang memperkenalkan jenis kesalahan tertentu. Mutasi bisa berbasis program, yang bisa saja di gunakan untuk memeriksa kemampuan set *test* untuk menemukan jenis kesalahan tertentu, atau berdasarkan spesifikasi secara sistematis menghasilkan kotak uji *black box*.

8. *Category Partition Method*

*Category Partition Method* digunakan untuk memformalkan dokumentasi kasus *black box* testing dalam bahasa bernama *Test Specification Language (TSL)*.

9. *Classification Trees*

Pada *Classification Trees* kategori dan pilihan diwakili dalam pohon, dengan kategori sebagai simpul akar, pilihan sebagai simpul daun dan simpul perantara mewakili dekomposisi kategori menjadi sub-kategori. Kasus uji dibangun dengan cara yang sama seperti CPM (yaitu dengan mengambil produk pilihan *Cartesian* dan menghapus pilihan yang tidak mungkin kombinasi), meskipun metode uji kombinatorial lainnya dapat digunakan untuk menentukan kasus uji. Kasus uji dicatat sebagai garis horizontal di bawah pohon yang bersinggungan dengan pilihan yang termasuk dalam setiap *test case*, dengan titik-titik hitam di persimpangan garis yang menandai pemilihan nilai data uji dimasukkan dalam kasus uji. (Murnane, 2009)

## Definisi

Menurut situs yang di tulis oleh Peni Kurniawati (2018) berjudul "Pengujian Sistem" menyebutkan bahwa *black box* testing atau sering disebut dengan pengujian fungsional adalah metode pengujian perangkat lunak yang di gunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program. (Kurniawati, 2018)

Menurut buku yang di tulis oleh S. Rao Vallabhaneni (2013) yang berjudul "*Wiley CIA Exam Review 2013: Internal Audit Knowledge Elements*" *black box* adalah metodologi uji dasar yang mengasumsikan tidak ada pengetahuan tentang struktur internal dan detail pelaksanaan dari objek penilaian. (Vallabhaneni, 2013)

## Perkembangan Teori

Pada penelitian yang di lakukan oleh Lila Setiyani pada jurnal yang berjudul “Pengujian Sistem Informasi *Inventory* Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing” mengatakan bahwa metode Black Box testing memberikan kemudahan untuk pengujian perangkat lunak untuk menguji *fungsinalitas* perangkat lunak. (Setiyani, 2019)

Pada penelitian yang di lakukan oleh Adi Krismadi, Ana Fisca, *dkk* pada jurnal yang berjudul “Pengujian Black Box berbasis *Equivalen Partitions* pada Aplikasi Seleksi Promosi Kenaikan Jabatan” mengatakan bahwa metode Black Box testing memberikan hasil pengujian yang sesuai antara perangkat lunak yang diuji dengan spesifikasi yang sudah di tentukan. (Krismadi et al., 2019)

## Daftar Pustaka

- Krismadi, A., Lestari, A. F., Pitriyah, A., Mardangga, I. W. P. A., Astuti, M., & Saifudin, A. (2019). Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 2(4), 155. <https://doi.org/10.32493/jtsi.v2i4.3771>
- Kurniawati, P. (2018, October 29). *Pengujian Sistem - SkyshiDigital - Medium*. <https://medium.com/skyshidigital/pengujian-sistem-52940ee98c77>
- Murnane, T. (2009). *The Atomic Rules Approach for Describing Black-Box Testing Methods and its Evaluation* [La Trobe University]. [http://homepage.cs.latrobe.edu.au/kreed/phd\\_students-completed\\_files/murnane.phd.thesis.2009.pdf](http://homepage.cs.latrobe.edu.au/kreed/phd_students-completed_files/murnane.phd.thesis.2009.pdf)
- Myers, G. J., Badgett, T., & Thomas with Corey Sandler, T. M. (2004). *The Art of Software Testing, Second Edition*. [www.Wiley.com](http://www.Wiley.com).
- Setiyani, L. (2019). Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing. *Techno Xplore: Jurnal Ilmu Komputer Dan Teknologi Informasi*, 4(1), 1–9. <https://doi.org/10.36805/technoxplore.v4i1.539>
- Vallabhaneni, S. R. (2013). *Wiley CIA Exam Review 2013, Internal Audit Knowledge Elements*. <https://books.google.co.id/books?id=f6BvL>

---

R8UQDMC&pg=PA500&dq=black+box+testing+adalah&hl=en&sa=X&ved=0ahUKEwi3k4Hx0IPoAhXUXSsKHcfVCTg4ChDoAQhqMAc#v=onepage&q=black box testing adalah&f=false

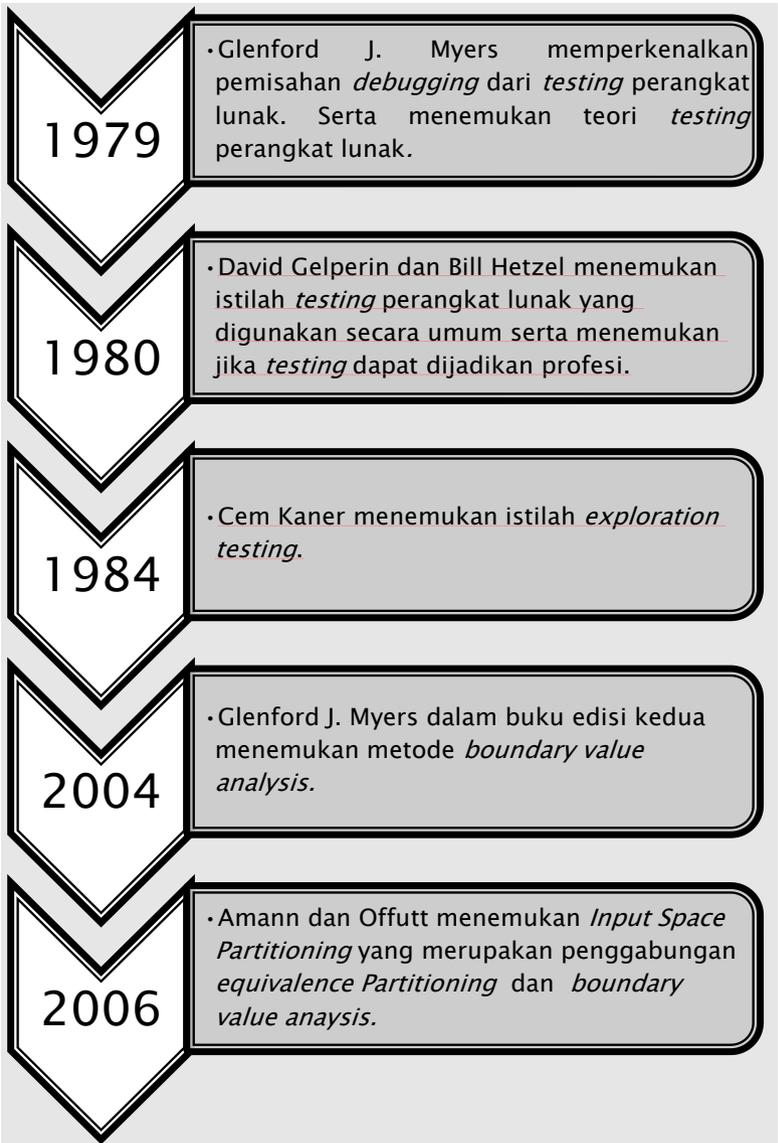
# Teori Boundary Value Analysis



**Kontributor :**

Made Putu Wijaya  
Vincentius Alfredo

- 321710005  
- 321710013



## **Pendahuluan**

Glenford J. Myers pertama kali mengemukakan pemisahan antara *debugging* dari testing untuk mendeteksi kesalahan dengan lebih mudah. Sehingga pada bukunya, beliau juga menemukan istilah testing dan penambahan *debugging* dengan verifikasi untuk pengembangan testing itu sendiri. Dimana dalam melakukan *testing* diperlukan kombinasi *input* yang banyak dan berbeda agar dapat lebih mudah mendeteksi kesalahan. (Myers, 1979)

Pada tahun berikutnya, Gelpherin dan Hetzel menemukan istilah *testing* pada perangkat lunak yang digunakan secara umum dan berpendapat jika *testing* perangkat lunak dapat dijadikan sebagai profesi. (Gelperin & Hetzel, 1980)

Pada tahun 1984, Cem Kaner menemukan istilah *exploration testing* yang inti dari teori ini adalah *tester* diberikan kebebasan dan tanggung jawab dalam melakukan *testing* untuk bisa mengoptimalkan proses dari perangkat lunak. (Kaner, 1984)

Glenford J. Myers dalam buku seri keduanya untuk *Art of Software Testing* pada tahun 2004 menemukan metode – metode dalam melakukan black box testing yaitu; Graph test, equivalence partitioning, serta boundary value analysis. (Myers, 2004)

Selanjutnya pada tahun 2006 Amann dan Offutt menemukan Input Space Partitioning yang merupakan perpaduan antara *equivalence partitioning* dan *boundary value analysis*. (Amann & Ofutt, 2006)

## Definisi

*Boundary Value Analysis* adalah Teknik yang berfokus dalam identifikasi representasi dari partisi yang sebisa mungkin paling dekat dengan batas (*boundaries*) partisi (Zander, Schieferdecker dan Mosterman, 2011).

*Boundary Value Analysis* menurut sumber lain juga merupakan sebuah pengembangan analisis posibilitas awal dan akhir untuk sebuah kelas *equivalence* (Everett dan McLeod, 2007).

Selain itu, *Boundary Value Analysis* juga berhubungan dekat dengan *equivalence partition*, dimana BVA akan lebih terkonsentrasi pada *value* dalam dalam kelas *equivalence* yang memiliki nilai *error* yang lebih besar (Ould dan Unwin, 1994).

## Perkembangan Teori

Dalam perkembangannya, *Boundary Value Analysis* telah digunakan dalam berbagai macam *testing*.

Dalam jurnal yang berjudul Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). BVA digunakan pada fungsi tambah kelas pada aplikasi prediksi kelulusan SNMPTN. Hasil pengujian menggunakan teori *Boundary Value Analysis* menunjukkan bahwa masih banyak kekurangan dalam proses validasi data sehingga dapat menyebabkan data yang dimasukan dapat tidak sesuai dengan keinginan. Sehingga dengan BVA, aplikasi yang ada dapat dikembangkan kembali (Mustaqbal, Firdaus dan Rahmadi, 2015).

Bukti penggunaan teori BVA dapat ditemukan pada jurnal *BWDM: Test Cases Automatic Generation Tool Based on Boundary Value Analysis with VDM++* . Dalam studi kasus tersebut dengan menggunakan *tool* yang telah dikembangkan, penggunaan BDWM (*Boundary Value/Vienna Development Method*) dapat

menghasilkan hasil *output* yang sesuai dengan spesifikasi VDM++. Namun, dengan BDWM penggunaan yang ada perlu penyesuaian lingkup agar sesuai dengan *requirement* yang diinginkan oleh pengguna *tool*. (Katayama et al., 2017)

## Daftar Pustaka

- Dobslaw, F., Gomes, F. and Feldt, R. (2020). Boundary Value Exploration for Software Analysis. arXiv. [online] Available at: <https://arxiv.org/abs/2001.06652> [Accessed 3 Mar. 2020].
- Guru99.com. (2020). Boundary Value Analysis & Equivalence Partitioning with Examples. [online] Available at: <https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html> [Accessed 3 Mar. 2020].
- Zander, J., Schieferdecker, I. and Mosterman, P. (2011). Model-Based Testing for Embedded Systems. CRC Press, p.93.
- Everett, G. and McLeod, R. (2007). Software testing. John Wiley & Sons, p.114.
- Ould, M. and Unwin, C. (1994). Testing in software development. Cambridge [Cambridgeshire]: Published by Cambridge University Press on behalf of the British Computer Society, p.80.
- Shi, M., 2010. Software Functional Testing from the Perspective of Business Practice. Computer and Information Science, 3(4).

- Mustaqbal, M., Firdaus, R. and Rahmadi, H., 2015. PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),.
- Katayama, T., Tachiyama, H., Kita, Y., Yamaba, H., Aburada, K. and Okazaki, N., 2017. BWDM: Test Cases Automatic Generation Tool Based on Boundary Value Analysis with VDM++. *Journal of Robotics, Networking and Artificial Life*, 4(2), p.110.

# Teori Equivalence Partitioning

**Kontributor :**

Farhan Adriansyah Ekadana - 321710003

Melvin Natalino Chandra - 321710007





## **Pendahuluan**

Berdasarkan dari buku *The Art of Software Testing*, oleh Glenford J. Myers pada tahun 1979 memperkenalkan pemisahan *debugging* dari *testing* yang bertujuan untuk mempermudah mendeteksi kesalahan pada perangkat lunak yang belum ditemukan. Sehingga muncul pengembangan dalam *testing* perangkat lunak yaitu dengan melakukan pemisahan misalnya *debugging* dengan verifikasi. Dalam melakukan *testing* harus memiliki banyak kombinasi inputan yang berbeda agar dapat mendeteksi kesalahan pada saat melakukan *testing* perangkat lunak. (Myers, 1979)

Pada tahun 1980, David Gelperin dan Bill Hetzel menemukan istilah *testing* perangkat lunak mulai banyak dikenal dan digunakan secara umum. Kemudian banyak orang yang berpandangan bahwa *testing* perangkat lunak dijadikan sebagai profesi. (Gelperin & Hetzel, 1980)

Berikutnya dari buku *A Tutorial in Exploratory Testing*, oleh Cem Kaner di tahun 1984 menemukan istilah *testing* dengan eksplorasi pada perangkat lunak dimana gaya terhadap *testing*

perangkat lunak berdasarkan kebebasan dan tanggung jawab penguji untuk mengoptimalkan dari *testing* kualitas perangkat lunak. (Kaner, 1984)

Selanjutnya pada tahun 1988, Balcer dan Ostrand membuat suatu metode *testing* perangkat lunak dengan cara membagi berdasarkan kategori *input testing* sesuai fungsinya. (Ostrand & Balcer, 1988)

Sedangkan pada tahun 2004, Badgett, Todd Thomas, dan Corey Sandler melakukan *testing* perangkat lunak dengan menggunakan metode *equivalence partitioning*. Metode *equivalence partitioning* ini memiliki cara yang akurat untuk mendeteksi kesalahan perangkat lunak. Dalam metode ini terdapat dua kategori input sesuai fungsinya yang dapat digunakan yaitu *it reduces* yang mendeskripsikan keperluan mengenai kebutuhan *testing* dengan jelas dan *it covers a large set of other possible test case* yang mendeskripsikan pemberitahuan mengenai kesalahan dalam perangkat lunak pada saat sedang melakukan *testing*. (Badgett et al., 2004)

## Definisi

*Testing* pada perangkat lunak ialah tahapan eksekusi dalam suatu perangkat lunak atau sistem yang melakukan kegiatan evaluasi atribut atau kemampuan dari suatu perangkat lunak atau sistem dan menghasilkan laporan pengujian perangkat lunak untuk perusahaan. (Perry, 2006)

*Blackbox testing* adalah teknik pada *testing* perangkat lunak dengan tujuan menguji spesifikasi fungsional perangkat lunak tersebut. Dalam penerapannya, teknik *blackbox testing* ini tidak melihat struktur kontrol dari perangkat lunak sehingga berfokus pada informasi di *domain* perangkat lunak. (Watkins & Mills, 2011)

*Blackbox testing equivalence partitioning* adalah teknik pada *blackbox testing* dengan tujuan mengurangi jumlah dari pengujian kasus. Dalam *equivalence partitioning* dilakukan pada *domain* yang dibagi seperti *input* program namun berdasarkan penentuan nilai *input*. (Pressman, 2006)

## **Perkembangan Teori**

Berdasarkan penelitian terdahulu pada jurnal yang berjudul "Pengujian *Black Box* berbasis *Equivalence Partitions* pada Aplikasi Seleksi Promosi kenaikan jabatan" yang ditulis oleh Adi Krismadi, Ana Fisca Lestari, Anisah Pitriyah, I Wayan Putra Ardhie Mardangga, Muji Astuti, dan Aris Saifudin berisi mengenai *testing* dengan menggunakan *black box* berbasis *equivalence partitions* terhadap aplikasi seleksi promosi dan kenaikan jabatan. Pengujian aplikasi seleksi promosi dan kenaikan jabatan ini dilakukan karena terdapat masalah jika melakukan penghapusan data. Dengan melakukan *testing* terhadap aplikasi seleksi promosi dan kenaikan jabatan dengan menggunakan *black box* berbasis *equivalence partitions* ditemukan adanya *bug* pada suatu *form* dari aplikasi seleksi promosi dan kenaikan jabatan. (Krismadi et al., 2019)

Berikutnya pada jurnal yang berjudul "Pengujian *Software* Aplikasi Perawatan Barang Milik Negara Menggunakan Metode *Black Box Testing Equivalence Partitions*" yang ditulis oleh Nur Wahyu Rahadi dan Cahya Vikasari berisi

mengenai *testing* terhadap *software* aplikasi perawatan barang milik negara dengan menggunakan metode *black box testing equivalence partitions*. *Testing* terhadap aplikasi perawatan milik negara ini akan dilakukan pada modul fungsional penggantian barang. Setelah melakukan pengujian terhadap aplikasi perawatan barang milik negara dengan menggunakan metode *black box testing equivalence partitions* diperoleh nilai tingkat efektivitas dari modul fungsional adalah sebesar 93,2%. (Rahadi & Vikasari, 2020)

Kedua penelitian tersebut menggunakan *black box testing equivalence partitioning* pada sebuah aplikasi fungsionalitas yang setelah dilakukan *testing* ditemukan *bug minor*. Oleh karena itu, metode *testing* ini dapat memberikan celah dalam aplikasi kepada *tester* dengan pembagian *domain input* ke beberapa kelas data sehingga *test case* dapat diperoleh. *Test case* ini dilakukan *testing* dengan *equivalence partitioning* dan hasil dari *testing* akan dilakukan perhitungan keefektifannya. Hal ini bertujuan untuk diperbaiki bagi *developer* dalam bentuk *maintenance* atau *patch update* pada kemudian hari.

## Daftar Pustaka

- Badgett, Thomas, T. M., & Sandler, C. (2004). The Art Of Software Testing. In *John Wiley & Sons, Inc* (Second). John Wiley & Sons, Inc. [www.ccscenet.org/cis](http://www.ccscenet.org/cis)
- Gelperin, D., & Hetzel, B. (1980). The Growth Of Software Testing. *Communication Of The ACM*, 31(6), 10. <https://doi.org/10.1145/62959.62965>
- Kaner, C. (1984). Exploratory Testing Professor of Software Engineering. In *Florida Institute of Technology* (First, Issue 1). Professor of Software Engineering. <http://www.kaner.com/pdfs/QAIExploring.pdf>
- Krismadi, A., Lestari, A. F., Pitriyah, A., Mardangga, I. W. P. A., Astuti, M., & Saifudin, A. (2019). Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 2(4), 155. <https://doi.org/10.32493/jtsi.v2i4.3771>
- Myers, G. J. (1979). The Art Of Software Testing. In *IEEE Proceedings of the National Aerospace and Electronics Conference* (First Edit, Vol. 2). John Wiley & Sons, Inc. <https://doi.org/10.1109/naecon.1991.165837>
- Ostrand, T. J., & Balcer, M. J. (1988). THE CATEGORY-PARTITION METHOD FOR SPECIFYING AND GENERATING FUNCTIONAL TESTS. *Communications of the ACM*, 31(6), 11.

- Perry, W. E. (2006). *Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates* (R. Elliott (ed.); Third). Wiley Publishing, Inc.
- Pressman, R. S. (2006). *Software Engineering A Practitioner's Approach* (F. M. Schilling (ed.); Seven). The McGraw-Hill Companies. <https://doi.org/10.1049/ic:20040411>
- Rahadi, N. W., & Vikasari, C. (2020). Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions. *Infotekmesin*, 11(1), 57–61. <https://doi.org/10.35970/infotekmesin.v11i1.124>
- Rubin, J., & Chisnel, D. (2008). Handbook of Usability Testing. In B. Elliot (Ed.), *Medicina Interna de Mexico* (Second, Vol. 17, Issue 2). Wiley Publishing, Inc.
- Watkins, J., & Mills, S. (2011). *Testing IT: An off-the-shelf software testing process* (L. Cowles & D. Jou (eds.); Second). Cambridge University Press.

# Studi Kasus Testing Yoopies

**Boundary Value Analysis**

**Kontributor :**

Gregorius Allan  
Ridho Wisnu F

321810006  
321810016



## **Pendahuluan**

Yoopies merupakan *platform* yang menyediakan layanan jasa pengasuh anak yang terdapat di Negara Perancis. *Platform* ini terbentuk pada bulan November 2011 yang didirikan oleh Benjamin Suchar. Yoopies memiliki tujuan yaitu menyediakan layanan jasa pengasuh anak nomor satu di Eropa dengan dasar jasa atas dasar kepercayaan konsumen. Target pasar dari *platform* ini adalah kalangan orang tua di daerah perkotaan. Perbedaan layanan ini dengan pesaingnya adalah kualitas pelayanan dan fasilitas dalam komunikasi yang baik antara orang tua dan pengasuh. Selain memberikan fasilitas yang baik dan mudah dipahami oleh orang tua, Yoopies juga melakukan wawancara, memeriksa ID, dan kualitas dari tiap pengasuh untuk mendapatkan pelayanan yang maksimal kepada konsumen.

<https://books.google.co.id/books?id=nDoIDwAAQBAJ&pg=PA53&lpg=PA53&dq=yoopies+childcare+founded&source=bl&ots=w45OxwtIia&sig=ACfU3U26p6qQXpbFwbCm5HhZkDGp5QJd4Q&hl=id&sa=X&ved=2ahUKEwiyyrHC1oPoAhWYbSsSKHd4VDA>

wQ6AEwAXoECAoQAQ#v=onepage&q=yoopies%20childcare%20founded&f=false

Yoopies menyediakan registrasi untuk mendaftar sebagai pengasuh dan konsumen. Untuk calon pengasuh dapat mengisi form pendaftaran dan dapat memperbarui data tersebut dengan tampilan yang mudah dipahami. Jika untuk konsumen dapat mendaftar pada bagian *sign up* dengan tampilan halaman yang mudah dimengerti. *Platform* ini juga memberikan tata cara mudah dalam proses pemesanan pengasuh, yaitu cukup dengan mengisi *postcode* dan alamat tempat tinggal. Hasilnya Yoopies memberikan informasi pengasuh yang tersedia dengan keterangan lengkap mengenai biodata, kemampuan, dan gaji. (Cyril, 2017)

Yoopies juga memberikan layanan pengasuh lainnya antara lain *tutoring*, *housekeeping*, *pet sitting*, dan *senior care*. Dengan jenis layanan yang diberikan diharapkan pelanggan dapat dengan mudah menemukan apa kebutuhannya. Selain itu Yoopies juga sudah dipercaya oleh banyak orang tua dalam urusan *home service*. Hal ini ditunjang dengan akuisisi *startup* Yokoro untuk memperluas *homecare*

*services*. Akuisisi ini bertujuan untuk memberikan ruang yang luas kepada pegawai atau pekerja Yoopies untuk mendapatkan pelanggan di daerah local mereka. (Boogar, 2014)

Maka dari itu pada pembahasan kali ini akan membahas tentang testing dengan metode *boundary value analysis* terhadap Yoopies, terutama dalam hal *babysitter*.

## **Pembahasan**

Pada *Yoopies* website dilakukan testing dengan desain dan tindakan desain yang dirancang dan dilakukan dengan metode *black box boundary analysis*. Metode ini menjalankan testing dengan melakukan pengujian terhadap nilai atau fungsi dengan memakai *range* nilai. Pemilihan nilai pengujian yang diambil disesuaikan dengan batas bawah, nilai sebenarnya, dan batas atas yang sudah diatur pada fungsi atau opsi pengisian pada *website* tersebut. (Guru99, n.d.)

Selain itu pengujian pada *website* ini diharapkan dapat memberikan hasil *testing* yang akurat dan jelas. Sehingga dapat mengetahui masalah yang ada di *website* tersebut untuk segera ditangani. Serta untuk memberikan performa yang baik kepada pelanggan berhubung *Yoopies* sudah dikenal dan dipakai oleh banyak orang

## Studi Kasus Testing Yoopies

---

<b>Test Case ID</b>	<b>Test Designed by :</b> Gregorius Allan
<b>Test Priority</b>	<b>Test Designed date:</b> 23 April 2020
<b>Module Name</b>	<b>Test Executed by :</b> Ridho Wisnu F
<b>Test Title</b> Penguian terhadap penginputan alamat awal registrasi di Yoopies	<b>Test Execution date :</b> Ridho Wisnu F
<b>Description</b>	
<b>Pre-Conditions</b>	User harus sudah menyiapkan beberapa data untuk pengisian form registrasi dan sudah berada di halaman registrasi alamat tempat tinggal
<b>Dependencies</b>	

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1.	Input nomor alamat dibawah nilai minimum	Nomor alamat : -1,-2,-3,..	Tampil alamat lain yang memiliki nilai terdek at dan bernilai positif	Tampil alamat lain yang memiliki nilai terdekat dan bernilai positif	Berhasil	

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
2.	Input nomor alamat minimum dan nilai dibawah minimum	Nomor alamat : 0 - 1, 0 - 4, ...	Tampil pilihan alamat di UK atau alamat lain yang memiliki nilai terdekat	Tampil pilihan alamat di UK atau alamat lain yang memiliki nilai terdekat	Berhasil	
3.	Input nomor alamat diatas maksimum	Nomor alamat : 1000 1,300 04, ...	Tidak menampilkan hasil nomor alamat	Tidak menampilkan hasil nomor alamat	Berhasil	
4.	Input nomor alamat sesuai dengan data yang	Nomor alamat : 32	Tampil daerah kota yang berawal nomor 32	Tampil daerah kota yang berawal nomor 32	Berhasil	

## Studi Kasus Testing Yoopies

---

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
	tersedia					

<b>Post-Conditions</b>	User dapat beralih ke halaman registrasi by email
------------------------	---

<b>Test Title</b> Pengujian terhadap penginputan data pada form registrasi Yoopies
<b>Description</b>
<b>Pre-Conditions</b> User harus sudah menyiapkan beberapa data untuk pengisian form registrasi dan sudah berada di halaman sign up by email
<b>Dependencies</b>

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
1.	Input karakter alpha betic yang berjumlah 5	Firstname : Ridho	Data firstname akan diterima oleh sistem	Data firstname akan diterima oleh sistem	Berhasil	

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
	karakter					
2.	Input karakter alpha betic yang berjumlah 5 karakter	Surname : Wisnu	Data surname akan diterima oleh sistem	Data surname akan diterima oleh sistem	Berhasil	
3.	Input alpha bet dan angka pada firtsname	Firstname : Ridho55	Firstname tidak dapat diterima oleh sistem dan muncul pesan invalid registrati on	Firsname tidak diterima oleh sistem dan muncul pesan "First name and last name must not contain special characters"	Berhasil	

## **Studi Kasus Testing Yoopies**

---

<b>No</b>	<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>	<b>Note</b>
4.	Input karakter spesial pada firstname	Firstname : @ridho	Firstname tidak akan diterima oleh sistem karena batasan karakter	Firstname tidak dapat diterima oleh sistem dan muncul notifikasi "First name and last name must not contain special characters"	Berhasil	Pada firstname tidak ada batasan inputan jumlah karakter
5.	Input alpha bet dan angka pada surname	Surname : Wisnu356434	Surname tidak dapat diterima oleh sistem dan muncul pesan invalid registration	Surname tidak diterima oleh sistem dan muncul pesan "First name and last name must not contain special characters"	Berhasil	
6.	Input karakter	Surname : @asdad	Username tidak akan	Surname tidak dapat diterima	Berhasil	Pada surname tidak ada batasan

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
	spesial pada surname		diterima oleh sistem karena batasan karakter	oleh sistem dan muncul notifikasi "First name and last name must not contain special characters"		inputan jumlah karakter
7.	Input karakter & dan angka minus pada password	Password : ridho&jin-987	Password tetap dapat diterima oleh sistem	Password tetap dapat diterima oleh sistem	Berhasil	Pada password menerima segala jenis karakter yang diinputkan
8.	Input karakter alpha betic saja pada	Password : ridhowi snufirm antya	Password ditolak sistem dan muncul pesan invalid	Password diterima oleh sistem dan dapat masuk ke halaman	Gagal	Password tidak mempunyai ketentuan karakter khusus

**Studi Kasus Testing Yoopies**

---

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
	password		registrasi	registrasi anak		

<b>Post-Conditions</b>	User dapat masuk ke halaman registrasi keperluan service babysitter (registrasi anak)
------------------------	---

<b>Test Title</b>	Pengujian terhadap penginputan data pada registrasi anak dan kebutuhannya
<b>Description</b>	
<b>Pre-Conditions</b>	User harus sudah menyiapkan beberapa data untuk pengisian form registrasi dan sudah berada di halaman registrasi identitas anak
<b>Dependencies</b>	

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
1.	Memilih jumlah anak dengan range angka 1	My children : 200 (x<3th n), 200	Data penentu an my children diterima	Data penentuan my children diterima oleh sistem	Berhasil	Semakin banyak jumlah anak, maka

**Studi Kasus Testing Yoopies**

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
	hingga >1000	( $3 < x < 6$ thn), dan 200 ( $x > 7$ th n)	oleh sistem			semakin turun performa dari website Yoopers
2.	Input karakter alphanumeric dan angka pada box ad title sebanyak 13 karakter	Ad title : Description	Ad title ditolak oleh sistem dan terdapat notifikasi himbauan untuk mentaati peraturan yang ada	Sistem menerima inputan Ad title dan tanpa ada notifikasi himbauan apapun	Gagal	Ketentuan panjang karakter adalah 30 - 60 karakter
3.	Input karakter alphanumeric sebanyak 30 karakter	Ad title : Description of my love 3000 family's	Ad title akan diterima oleh sistem	Ad title akan diterima oleh sistem	Berhasil	

## **Studi Kasus Testing Yoopies**

---

<b>No</b>	<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>	<b>Note</b>
4.	Input karakter alphabetic dan angka sebanyak 23 karakter .	Define your needs : kami sedang membutuhkan 12	Data define your needs akan diterima oleh sistem	Sistem menolak define your needs dan keluar pesan harus menuliskan sebanyak 30 karakter atau lebih	Gagal	Pada box define your needs tidak ada himbauan tentang minimal dan maksimal karakter yang dapat diinputkan
5.	Inputkan karakter alphabetic dan angka lebih dari 30 karakter	Define your needs : kami sedang membutuhkan 12 pengasuh	Data define your needs akan diterima oleh sistem	Data define your needs akan diterima oleh sistem	Berhasil	
6.	Input nomor telepon dengan range	Nomor telepon : 235139 41290	Nomor telepon diterima sistem dan tidak	Nomor telepon diterima oleh sistem dan tidak	Berhasil	

## Studi Kasus Testing Yoopies

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
	angka 0 - 9		muncul notifikasi apapun	muncul notifikasi apapun		

<b>Post-Conditions</b>	User dapat masuk ke halaman beranda atau halaman utama Yoopies
------------------------	--

<b>Test Title</b> Pengujian terhadap penginputan data pada booking babysitter
<b>Description</b>
<b>Pre-Conditions</b> User harus sudah masuk di halaman booking babysit
<b>Dependencies</b>

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
1.	Input karakter angka dengan settingan waktu	hours per week : 46 hours	Waktu 46 hours ditolak oleh sistem	Waktu 46 hours ditolak oleh sistem dan menampilkan notifikasi	Berhasil	

**Studi Kasus Testing Yoopies**

---

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
	diatas 45 hours		dan mena mpilka n notifik asi data tidak dikena li	"sorry, no matching option"		
2.	Memilih opsi pilihan waktu yang sama pada waktu from dan to	From : 00:00 dan To : 00:00	Waktu From : 00:00 dan To : 00:00 ditolak oleh sistem dan muncu l notifik asi invalid	Waktu From : 00:00 dan To : 00:00 ditolak oleh sistem dan muncul notifikasi "Schedule is not valid, please adjust your reservation "	Berhasil	

No	Test Steps	Test Data	Expected Result	Actual Result	Status	Note
3.	Memilih opsi pilihan waktu from dan to yang sudah disediakan	From : 00:00 dan To : 01.30	Waktu From : 00:00 dan To : 01:00 diterima oleh sistem	Waktu From : 00:00 dan To : 01:00 diterima oleh sistem	Berhasil	
4.	Input tanggal end date bab work melewati tanggal 23 April 2021	End date : 24 April 2021	Tanggal ditolak oleh sistem dan memberikan pilihan tanggal yang berakhir pada	Tanggal ditolak oleh sistem dan memberikan opsi pilihan tanggal yang berakhir pada tahun 2021	Berhasil	

## **Studi Kasus Testing Yoopies**

---

<b>No</b>	<b>Test Steps</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>	<b>Note</b>
			tahun 2021			
5.	Input tanggal end date babysitter work pada tahun 2021	End date : 01 April 2021	Tanggal end date : 01 April 2021 diterima oleh sistem	Tanggal end date : 01 April 2021 diterima oleh sistem	Berhasil	
6.						

<b>Post-Conditions</b>	User dapat lanjut ke halaman payment untuk pesan babysitter
------------------------	---

## **Kesimpulan**

Menurut kelompok kami, website Yoopies telah memenuhi standar dan layak untuk digunakan. Website yang mudah diakses dan tampilan yang friendly mudah digunakan untuk semua orang. Hal ini didukung oleh website itu sendiri yang telah digunakan untuk melayani jutaan orang dari berbagai negara dengan menyediakan data babysiter yang akurat dan disesuaikan dengan kebutuhan pelanggan. Tidak hanya itu, website Yoopies juga memiliki filter berbagai macam bahasa untuk mempermudah berbagai kalangan masyarakat dari seluruh penjuru dunia untuk menikmati pelayanan Yoopies.

Namun, pada website Yoopies ini sendiri tidak luput dari kelemahan, yaitu minimnya informasi yang disediakan untuk mengisi form. Berbeda dengan kelengkapan informasi Yoopies dalam menjelaskan pelayanan mereka, form yang mereka sediakan untuk mengisi identitas babysiter dan pengguna layanan Yoopies seakan tidak dimaksimalkan dengan sepenuhnya.

Secara keseluruhan website dari Yoopies ini sudah memenuhi kebutuhan dan dapat menjalankan proses bisnis mereka dengan akurat dan maksimal. Website yang dapat diakses melalui berbagai platform, user interface yang lengkap dan mudah digunakan, serta pelayanan cepat dengan data akurat menjadi kunci berhasilnya website Yoopies.

## Daftar Pustaka

- Boogar, L., 2014. *European babysitter marketplace Yoopies acquires Yokoro to expand into homecare.* [Online]  
Available at:  
<https://yoopies.fr/presse/yoopies-acquires-yokoro-childcare>  
[Accessed 23 April 2020].
- Cyril, B., 2017. *The Old, The New and the Future Yoopies.* [Online]  
Available at:  
<https://medium.com/xangevc/the-old-the-new-and-the-future-yoopies-79e4e14127df>  
[Accessed 23 April 2020].
- Guru99, n.d. *Boundary Value Analysis & Equivalence Partitioning with Examples.* [Online]  
Available at:  
<https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html>  
[Accessed 23 April 2020].



---

Berbahagialah, bukan karena segala sesuatu baik, tetapi karena kamu mampu melihat hal baik dari segala sesuatu.

---





# **Studi Kasus Testing Qasir**

## **Boundary Value Analysis**

**Kontributor :**

Jose Fernando Wijaya  
Anis Setyaningrum

- 321810008  
- 321810021

## **Pendahuluan**

Dikutip dari [youngster.id](https://youngster.id), Qasir telah sukses membantu UMKM dalam menjalankan bisnisnya dengan platform Point Of Sales (POS). Hingga kini aplikasi Qasir telah digunakan di 500 kota Indonesia.

Pada akhir tahun 2019, Qasir telah mencatatkan nilai transaksi sebesar Rp1,5 Triliun, sekitar 0,2% total pergerakan ekonomi Indonesia. Kedepannya, Michael Johan Williem, CEO Qasir, menargetkan kenaikan pengguna Qasir sebanyak 300.000 user.

<https://youngster.id/news/pos-qasir-hadir-di-500-kota-untuk-kembangkan-bisnis-umkm/>

Dilansir dari [Kabar Jatim.com](https://kabarjatim.com) yang diakses pada tanggal 5 Maret 2020 dengan [link https://kabarjatim.com/punya-warung-dua-aplikasi-ini-patut-dicoba/](https://kabarjatim.com/punya-warung-dua-aplikasi-ini-patut-dicoba/), Salah satu kesalahan utama pemilik UMKM dalam mengelola usaha sehingga tetap stagnan adalah karena tidak adanya pemahaman dasar soal pengelolaan bisnis. Karenanya, penggunaan platform POS bagi bisnis UMKM diharapkan mampu memangkas risiko

human error dalam proses pengelolaan usaha ujar Michael Williem selaku CEO Qasir.

Dilansir dari Medcom.ID yang diakses pada tanggal 5 Maret 2020 dengan *link* <https://www.qasir.id/studi-kasus/gimana-sih-zaskia-mecca-mengatur-kesibukan-sambil-berdagang>,

Zaskia Mecca artis Indonesia untuk mengelola bisnis clothing linenya, ia mempercayakannya dengan menggunakan aplikasi Qasir dalam mencatat penjualan dan stok baju. Sebelum menggunakan aplikasi Qasir, untuk penjualan di store Meccanism pencatatan masih dilakukan manual di nota lalu harus direkap kembali ke laptop sehingga membutuhkan waktu yang banyak. Aplikasi Qasir gratis, sangat mudah digunakan dan saya tinggal memonitoring dari web saja yang update terus.

## Testing

Project Name	Test Qasir
Release version	3.32.2-build.1
Test Designed by	Jose Fernando and Anis Setyaningrum
Test Designed Date	April 23, 2020
Test Executed By	Jose Fernando and Anis Setyaningrum
Test Execution Date	April 23, 2020

Configuration Testing	
Hardware	Smartphone vivo s1 pro Qualcomm Snapdragon 665 2.02 GHz Ram 8 Gb
Software	Android 9.0

Test Case	Q1
Test Title	Membuka aplikasi Qasir
Description	Memeriksa aksesibilitas aplikasi
Pre-conditions	Telah memiliki aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Aplikasi terbuka

Step	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan aplikasi Qasir	Tekan	Aplikasi terbuka	As Expected	Pass



## Studi Kasus Testing Qasir

---

Test Case	Q2
Test Title	Registrasi di aplikasi Qasir
Description	Memeriksa fitur registrasi
Pre-conditions	Telah membuka aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Berhasil membuat akun

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan tombol Daftar Sekarang	tekan	Menampilkan halaman pendaftaran	As Expected	Pass
2	Mengisi kolom Nama Pemilik	Jose Fernando	Muncul data yang dimasukkan	As Expected	Pass
3	Mengisi kolom Email (opsional)	Email asli	Muncul data yang dimasukkan	As Expected	Pass
4	Mengisi kolom No handphone	Nomor 11 atau 12 digit asli	Muncul data yang dimasukkan	As Expected	Pass
5	Mengisi kolom Password 6 angka	Password 6 angka	Muncul 6 lambang titik	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
6	Menekan tombol Lanjut	Tekan	Muncul halaman pendaftaran berikutnya	As Expected	Pass
7	Mengisi kolom Nama Usaha	Toko Wijaya	Muncul data yang dimasukkan	As Expected	Pass
8	Menekan dropdown Pilih Jenis Usaha	Memilih Mini Market / Kelontong / Retail	Muncul data yang dipilih	As Expected	Pass
9	Menekan Pilih Kelurahan	Tekan	Muncul popup Cari Kelurahan	As Expected	Pass
10	Mengisi dan memilih kelurahan	Tasikmadu, Lowokwaru, Malang	Muncul pilihan terpilih dan tombol simpan dapat ditekan	As Expected	Pass
11	Menekan Simpan	Tekan	Pop up tertutup dan muncul data yang dipilih	As Expected	Pass
12	Menekan tombol	Tekan	Dapat mengubah	As Expected	Pass

## **Studi Kasus Testing Qasir**

---

No	Test Step	Test Data	Expected result	Actual Result	Status
	pensil untuk mengubah alamat web usaha		nama alamat web		
13	Mengisi kolom alamat web Nama Usaha	tokowija ya123	Muncul data yang dimasukkan	As Expected	Pass
14	Menekan tombol Lanjut	Tekan	Muncul halaman verifikasi	As Expected	Pass
15	Memasukkan nomor verifikasi	Nomor verifikasi	Muncul halaman utama aplikasi	As Expected	Pass

[← Daftar](#)

Masukan Data Diri Anda

Nama Pemilik

Email (opsional)

No handphone

Password 6 angka

000000 

Lanjut

## **Studi Kasus Testing Qasir**

---

Test Case	Q3
Test Title	Registrasi di aplikasi Qasir dengan data yang salah
Description	Memeriksa fitur registrasi
Pre-conditions	Telah membuka aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Berhasil membuat akun

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan tombol Daftar Sekarang	Tekan	Menampilkan halaman pendaftaran	As Expected	Pass
2	Mengisi kolom Nama Pemilik dengan	sdsdsd123@@@	Muncul data yang dimasukkan	As Expected	Pass
3	Mengisi kolom Email salah (opsional)	Asa (email tidak lengkap)	Muncul Format email salah	As Expected	Pass
4	Mengisi kolom No handphone salah	081 (kurang dari 10 digit)	Muncul Format No handphone tidak valid	As Expected	Pass
5	Mengisi kolom Password	222	Muncul Format password	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
	kurang dari 6 angka		harus 6 angka		
6	Menekan tombol Lanjut	Tekan	Tombol berwarna abu-abu dan tidak dapat ditekan	As Expected	Pass
7	Mengisi kolom Nama Usaha	aaa	Muncul Nama usaha minimal 5 huruf	As Expected	Pass
8	Tidak menekan dropdown Pilih Jenis Usaha	Tidak memilih	Tidak muncul apa - apa	As Expected	Pass
9	Menekan Pilih Kelurahan	Tekan	Muncul popup Cari Kelurahan	As Expected	Pass
10	Mengisi dan memilih kelurahan yang salah	Tadikmadu	Tidak muncul pilihan	As Expected	Pass
11	Mengisi kolom alamat web Nama Usaha	toko-wijaya	Muncul Nama Toko sudah ada sebelumnya	As Expected	Pass
11.1	Mengisi kolom	tok	Muncul Website toko	As Expected	Pass

## **Studi Kasus Testing Qasir**

---

No	Test Step	Test Data	Expected result	Actual Result	Status
	alamat web Nama Usaha		minimal 5 huruf		
12	Menekan tombol Lanjut	Tekan	Tombol berwarna abu-abu dan tidak dapat ditekan	As Expected	Pass
13	Memasukkan nomor verifikasi yang salah	1111	Muncul Kode aktivasi tidak valid	As Expected	Pass

← Daftar

### Masukan Data Diri Anda

Nama Pemilik  
sdsdsd1

Email (opsional)  
Asa  
Format email salah. Contoh: budi@gmail.com

No handphone  
081  
Format No handphone tidak valid.

Password 6 angka  
● ● ● | 🔒  
Format password harus 6 angka.

Lanjut

← Daftar

### Masukan Data Usaha Anda

Nama Usaha  
aaa  
Nama Usaha minimal 5 huruf.

Pilih Jenis Usaha

\*wajib diisi

Pilih Kelurahan

Nama Usaha  
toko-wijaya ✖ .qasir.id ✎  
Nama Toko sudah ada sebelumnya.  
Ini akan menjadi alamat web toko kamu.

Lanjut

← Daftar

### Masukan Data Usaha Anda

Nama Usaha  
aaa  
Nama Usaha minimal 5 huruf.

Pilih Jenis Usaha

\*wajib diisi

Pilih Kelurahan

Nama Usaha  
tok ✖ .qasir.id ✎  
Website toko minimal 5 huruf.  
Ini akan menjadi alamat web toko kamu.

Lanjut

← Daftar

### Masukan Data Usaha Anda

tadikmadu ✖

BATAL SIMPAN

Lanjut

## Studi Kasus Testing Qasir

---

Test Case	Q4
Test Title	Login di aplikasi Qasir
Description	Memeriksa fitur login dengan akun asli
Pre-conditions	Telah membuka aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Berhasil membuat akun

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan tombol Masuk	Tekan	Menampilkan halaman login	As Expected	Pass
2	Mengisi kolom No.HP / Email / ID Toko	Nomor HP	Muncul data yang dimasukkan	As Expected	Pass
2.1	Mengisi kolom No.HP / Email / ID Toko	Email asli	Muncul data yang dimasukkan	As Expected	Pass
2.2	Mengisi kolom No.HP / Email / ID Toko	ID Toko	Muncul data yang dimasukkan	As Expected	Pass
3	Mengisi kolom	Password	Muncul 6 lambang titik dan muncul	As Expected	Pass

## ***Studi Kasus Testing Qasir***

No	Test Step	Test Data	Expected result	Actual Result	Status
	Password 6 angka		halaman utama		

←

**Qasir**

No. HP / Email / ID Toko

No PIN 

[Lupa PIN?](#)

Versi 3.32.2-build.1

Test Case	Q5
Test Title	Login di aplikasi Qasir dengan data yang salah
Description	Memeriksa fitur login
Pre-conditions	Telah membuka aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Berhasil membuat akun

## Studi Kasus Testing Qasir

---

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan tombol Masuk	Tekan	Menampilkan halaman login	As Expected	Pass
2	Mengisi kolom No.HP / Email / ID Toko	081	Muncul data yang dimasukkan	As Expected	Pass
2.1	Mengisi kolom Password 6 angka	Password	Muncul No.hp atau Email harus minimal 5 karakter	As Expected	Pass
3.1	Mengisi kolom No.HP / Email / ID Toko	Abcdef	Muncul data yang dimasukkan	As Expected	Pass
3.2	Mengisi kolom Password 6 angka	Password	Muncul Silahkan login menggunakan No.HP / Email	As Expected	Pass
4.1	Mengisi kolom No.HP / Email / ID Toko	Email yang belum didaftarkan	Muncul data yang dimasukkan	As Expected	Pass

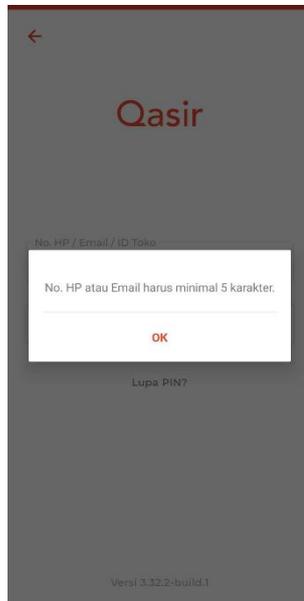
---

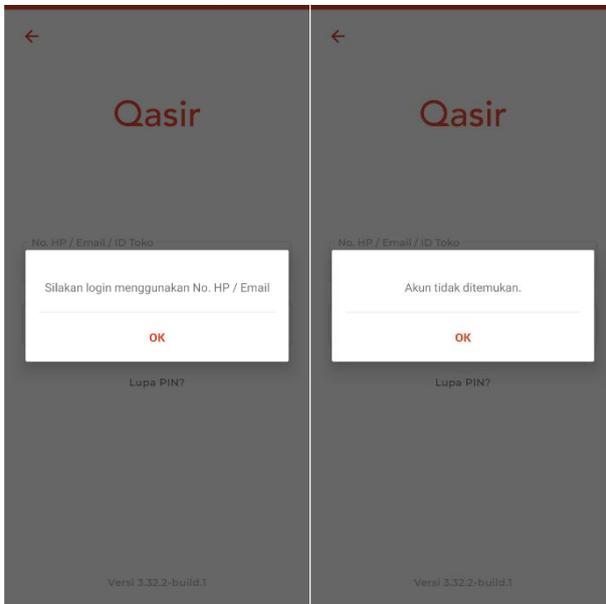
**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
4 .2	Mengisi kolom Password 6 angka	Password	Muncul Akun tidak ditemukan	As Expected	Pass
5 .1	Mengisi kolom No.HP / Email / ID Toko	Email asli	Muncul data yang dimasukkan	As Expected	Pass
5 .2	Mengisi kolom Password 6 angka	Password yang salah	Muncul Password kamu salah	As Expected	Pass

## Studi Kasus Testing Qasir

---

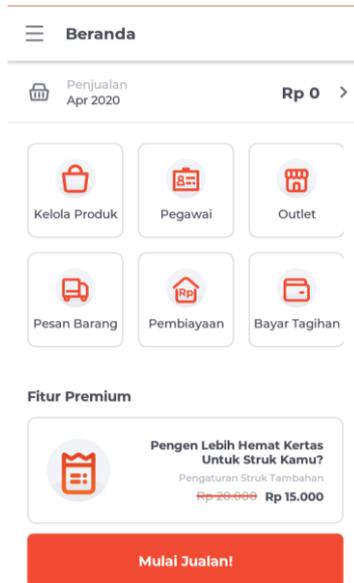




## Studi Kasus Testing Qasir

---

Test Case	Q6
Test Title	Masuk pada Beranda aplikasi Qasir
Description	Memeriksa fitur Beranda
Pre-conditions	Berhasil masuk dengan login
Dependencies	Memiliki koneksi internet
Post conditions	Telah masuk dalam beranda aplikasi Qasir



Test Case	Q7
Test Title	Menginput produk di aplikasi Qasir
Description	Memeriksa fitur 'Kelola Produk'
Pre-conditions	Berada di Beranda aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Telah masuk dalam fitur 'Kelola Produk'

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan tombol Tambah Produk	Tekan	Menampilkan halaman kelola produk	As Expected	Pass
1.1	Menekan icon camera	Tekan	Menampilkan pilihan galeri dan camera	As Expected	Pass
2	Mengisi kolom Nama Produk	Sunny	Muncul data yang dimasukkan	As Expected	Pass
3	Memilih Merk	Tekan	Muncul pilihan Merk	As Expected	Pass

## **Studi Kasus Testing Qasir**

---

No	Test Step	Test Data	Expected result	Actual Result	Status
4	Menekan tombol 'Tambah Merk'	Tekan	Menampilkan pop up form	As Expected	Pass
5	Mengisi kolom nama merk	So Nice	Muncul data yang dimasukkan	As Expected	Pass
6	Menekan tombol simpan	Tekan	Tidak muncul notifikasi	As Expected	Pass
7	Memilih kategori	Tekan	Muncul pilihan kategori	As Expected	Pass
8	Menekan tombol 'Tambah Kategori'	Tekan	Menampilkan pop up form	As Expected	Pass
9	Mengisi kolom nama kategori	Snack	Muncul data yang dimasukkan	As Expected	Pass
10	Menekan tombol simpan	Tekan	Tidak muncul notifikasi	As Expected	Pass
11	Mengisi kolom harga jual	Rp. 25200	Muncul data yang	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
			dimasukkan		
12	Menggeser icon tambah deskripsi	Geser ke kanan	Muncul form pendaftaran berikutnya	As Expected	Pass
13	Mengisi kolom harga modal	Rp. 22500	Muncul data yang dimasukkan	As Expected	Pass
14	Mengisi kode produk/barcode	Scan barcode kemasan	Muncul data yang dimasukkan	As Expected	Pass
15	Milih satuan unit	Tekan	Muncul pop up pilihan dan tambah satuan unit	As Expected	Pass
16	Mengisi stok (optional)	Input dengan angka	Muncul data yang dimasukkan	As Expected	Pass

## **Studi Kasus Testing Qasir**

---

No	Test Step	Test Data	Expected result	Actual Result	Status
17	Mengisi minimum stok	Input dengan angka	Muncul data yang dimasukkan	As Expected	Pass
18	Centang pada kolom untuk memberikan notifikasi saat stok mencapai batas minimum	Tekan	Kolom tercentang	As Expected	Pass
19	Menekan tombol 'Tambah Varian'	Tekan	Menampilkan halaman form variasi produk	As Expected	Pass
20	Mengisi kolom nama variasi	Nama produk	Muncul data yang dimasukkan	As Expected	Pass
21	Mengisi harga jual	Harga jual	Muncul data yang	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
			dimasukkan		
22	Menggeser icon tambah deskripsi	Geser ke kanan	Muncul form pendaftaran berikutnya	As Expected	Pass
23	Mengisi kolom harga modal	Input dengan angka	Muncul data yang dimasukkan	As Expected	Pass
24	Mengisi kode produk/barcode	Scan barcode kemasan	Muncul data yang dimasukkan	As Expected	Pass
25	Mimilih satuan unit	Tekan	Muncul pop up pilihan dan tambah satuan unit	As Expected	Pass
26	Menekan tombol 'Tambah Satuan Unit'	Tekan	Menampilkan pop up form	As Expected	Pass

## **Studi Kasus Testing Qasir**

---

No	Test Step	Test Data	Expected result	Actual Result	Status
27	Mengisi kolom nama satuan	Input dengan angka dan huruf	Muncul data yang dimasukkan	As Expected	Pass
28	Menekan tombol simpan	Tekan	Tidak muncul notifikasi	As Expected	Pass
29	Mengisi stok (optional)	Jumlah stok digudang	Muncul data yang dimasukkan	As Expected	Pass
30	Mengisi minimum stok	Input dengan angka	Muncul data yang dimasukkan	As Expected	Pass
31	Centang pada kolom untuk memberikan notifikasi saat stok mencapai batas minimum	Tekan	Kolom tercentang	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
32	Menekan tombol simpan	Tekan	Menampilkan halaman kelola produk	As Expected	Pass
33	Menekan tombol simpan	Tekan	Menampilkan halaman pencarian produk	As Expected	Pass

## Studi Kasus Testing Qasir

---

← Kelola Produk



Nama Produk  
Sunny

Merik  
So nice

Kategori  
snack

Varian

Re	Reguler Rp 25.200	6980 P...	>
Yu	yul	80 Pcs	>

← Kelola Produk

Nama Produk  
Sunny

Merik  
Snack

Nama Merek

BATAL SIMPAN

Tambah Deskripsi?

Tambah Varian

Simpan

### ← Variasi Produk

Nama Variasi  
Reguler

Harga Jual  
Rp 25.200

Tambah Deskripsi?

Harga Modal  
Rp 22.500

Kode Produk / Barcode  
hjiu12

Satuan Unit : Pcs >

Stok (tidak wajib diisi)  
6980

Minimum Stok  
1000

### ← Variasi Produk

Nama Variasi  
yul

Harga Jual  
Rp 200

Tambah Deskripsi?

Harga Modal  
Rp 180

Kode Produk / Barcode  
0lhd0

Satuan Unit : Pcs >

Stok (tidak wajib diisi)  
80

Minimum Stok  
15

## Studi Kasus Testing Qasir

---

Test Case	Q8
Test Title	Mengisi produk dagangan di aplikasi Qasir dengan data yang salah
Description	Memeriksa fitur 'Kelola Produk'
Pre-conditions	Berada di Beranda aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Telah masuk dalam fitur 'Kelola Produk'

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan tombol Tambah Produk	Tekan	Menampilkan halaman kelola produk	As Expected	Pass
1.1	Menekan icon camera	Tekan	Menampilkan pilihan galeri dan camera	As Expected	Pass
2	Mengisi kolom Nama Produk	rgfgjhjfj	Muncul data yang dimasukkan	As Expected	Fail
3	Memilih Merk	Tekan	Muncul pilihan Merk	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
4	Menekan tombol 'Tambah Merk'	Tekan	Menampilkan pop up form	As Expected	Pass
5	Mengisi kolom nama merk	hagahhhaviwq	Muncul data yang dimasukkan	As Expected	fail
6	Menekan tombol simpan	Tekan	Tidak muncul notifikasi	Tidak muncul peringatan	Fail
7	Memilih kategori	Tekan	Muncul pilihan kategori	As Expected	Pass
8	Menekan tombol 'Tambah Kategori'	Tekan	Menampilkan pop up form	As Expected	Pass
9	Mengisi kolom nama kategori	hfnk	Muncul data yang dimasukkan	As Expected	Fail
10	Menekan tombol simpan	Tekan	Tidak muncul	As Expected	Pass

## Studi Kasus Testing Qasir

---

No	Test Step	Test Data	Expected result	Actual Result	Status
			notifikasi		
11	Mengisi kolom harga jual	Rp. 092728760	Muncul kolom kosong	Muncul peringatan 'Harga melebihi batas Rp. 999,999,999	Fail
12	Menggeser icon tambah deskripsi	Geser ke kanan	Muncul form pendaftaran berikutnya	As Expected	Pass
13	Mengisi kolom harga modal	Rp. 092728760	Muncul data yang dimasukkan	Tidak muncul peringatan	Fail
14	Mengisi kode produk/bar code	Mengisi dengan angka acak	Muncul data yang dimasukkan	As Expected	Fail
15	Mimilih satuan unit	Tekan	Muncul pop up pilihan	As Expected	Pass

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
			dan tambah satuan unit		
16	Mengisi stok (optional)	Mengisi dengan angka acak	Muncul data yang dimasukkan	As Expected	Fail
17	Mengisi minimum stok	Mengisi dengan angka dan huruf acak	Muncul data yang dimasukkan	As Expected	Fail
18	Centang pada kolom untuk memberikan notifikasi saat stok mencapai batas minimum	Tekan	Kolom tercentang	As Expected	Pass
19	Menekan tombol 'Tambah Varian'	Tekan	Menampilkan halaman form	As Expected	Pass

## Studi Kasus Testing Qasir

---

No	Test Step	Test Data	Expected result	Actual Result	Status
			variasi produk		
20	Mengisi kolom nama variasi	jdhjbhkdjh ahgahsb	Muncul data yang dimasukkan	As Expected	Fail
21	Mengisi harga jual	Mengisi dengan angka acak lebih dari 9 angka	Muncul data yang dimasukkan	Muncul kolom kosong	Fail
22	Menggeser icon tambah deskripsi	Geser ke kanan	Muncul form pendaftaran berikutnya	As Expected	Pass
23	Mengisi kolom harga modal	Mengisi dengan angka acak lebih dari 9 angka	Muncul data yang dimasukkan	Muncul kolom kosong	Fail
24	Mengisi kode produk/bar code	Mengisi dengan angka acak	Muncul data yang dimasukkan	As Expected	Fail

---

**Studi Kasus Testing Qasir**

No	Test Step	Test Data	Expected result	Actual Result	Status
25	Mimilih satuan unit	Tekan	Muncul pop up pilihan dan tambah satuan unit	As Expected	Pass
26	Menekan tombol 'Tambah Satuan Unit'	Tekan	Menampilkan pop up form	As Expected	Pass
27	Mengisi kolom nama satuan	jdhjbkdjha hgahsb	Muncul data yang dimasukkan	As Expected	Fail
28	Menekan tombol simpan	Tekan	Tidak muncul notifikasi	As Expected	Pass
29	Mengisi stok (optional)	Mengisi dengan angka acak	Muncul data yang dimasukkan	As Expected	Fail

## **Studi Kasus Testing Qasir**

---

No	Test Step	Test Data	Expected result	Actual Result	Status
30	Mengisi minimum stok	Mengisi dengan angka acak	Muncul data yang dimasukkan	As Expected	Fail
31	Centang pada kolom untuk memberikan notifikasi saat stok mencapai batas minimum	Tekan	Kolom tercentang	As Expected	Pass
32	Menekan tombol simpan	Tekan	Menampilkan halaman kelola produk	As Expected	Pass
33	Menekan tombol simpan	Tekan	Menampilkan halaman pencarian produk	As Expected	Pass

### ← Kelola Produk

NP



Name Produk  
rgfghjffjui

Merik  
hagahhhaviwg >

Kategori  
hfnc Harga melebihi batas Rp. 999,999,999

Harga Jual

---

Tambah Deskripsi?

Tambah Varian

### ← Variasi Produk

Name Variasi  
Reguler 

Harga Jual  
Rp 0

---

Tambah Deskripsi?

Harga Modal

Kode Produk / Barcode  
t66gghúoookkjhvgjijj 

Satuan Unit : jdjhbkdjhahgahsb >

Stok (tidak wajib diisi)  
999999999.99

Minimum Stok  
64846

## Studi Kasus Testing Qasir

---

### ← Variasi Produk

Nama Variasi

gabajabajb

Harga Jual

Rp 56.480

Tambah Deskripsi?

Harga Modal

Rp 5.673

Kode Produk / Barcode

hsvsibajav

Satuan Unit : Pcs

Stok (tidak wajib diisi)

6945480688

Minimum Stok

8707

### ← Kelola Produk

Rg

Nama Produk

rgfghjfu

Merk

hagahhaviwg

Kategori

hfnk

Varian

Re	Reguler Rp 0	999999... >
ꦒꦧ	gabajabajb Rp 56.480	694548... >

## ***Studi Kasus Testing Qasir***

Test Case	Q9
Test Title	Mengisi data pegawai dengan data salah
Description	Memeriksa fitur 'Pegawai'
Pre-conditions	Berada di Beranda aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Telah masuk dalam fitur 'Tambah Pegawai'

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan fitur pegawai	Tekan	Menampilk an halaman pecarian	As Expected	Pass
2	Menekan symbol '+'	Tekan	Muncul form data pegawai	As Expected	Pass
3	Mengisi kolom Nama Pegawai	Nama huruf acak	Muncul data yang dimasukka n	As Expected	Fail
4	Menguploa d foto pegawai	Foto	Muncul foto yang telah diupload	As Expected	Pass
5	Mengisi kolom jabatan	jahahuabdb	Muncul data yang dimasukka n	As Expected	Fail
6	Memilih hak akses	Tekan	Muncul pilihan hak akses	As Expected	Pass

## ***Studi Kasus Testing Qasir***

---

No	Test Step	Test Data	Expected result	Actual Result	Status
7	Mengisi kolom no. handphone (optional)	Nama angka acak	Muncul data yang dimasukkan	As Expected	Fail
8	Mengisi kolom email	Nama huruf acak	Muncul data yang dimasukkan	Muncul pemberitahuan format salah dan contoh format yang benar	Fail
9	Mengisi password	password lebih dari 6 digit	Data tidak muncul	As Expected	Fail
10	Menekan tombol pilih outlet	Tekan	Menampilkan pilihan outlet	As Expected	pass
11	Menekan tombol simpan	Tekan	Menampilkan pada form data pegawai	As Expected	pass
12	Menekan tombol simpan	Tekan	menampilkan pop up keberhasilan	pop up pemberitahuan 'email harus berupa alamat surel yang	Fail

---

***Studi Kasus Testing Qasir***

No	Test Step	Test Data	Expected result	Actual Result	Status
				valid. Daftar outlet wajib diisi.	

## Studi Kasus Testing Qasir

---

← **Tambah Pegawai** **SIMPAN**



Nama Pegawai  
japappkvjavvjwhi

**Detail Pegawai**

Jabatan  
jahahuabdb

Hak Akses **Supervisor** ▾

No. Handphone (Wajib Diisi)  
8404546364948

Alamat Email  
haoahsvsvh  
Format email salah. Contoh: budi@gmail.com

Password 6 angka  
**085523** 

← **Tambah Pegawai** **SIMPAN**

**Detail Pegawai**

Jabatan  
jahahuabdb

Hak Akses **Supervisor** ▾

Email harus berupa alamat surel yang valid.  
Daftar Outlet wajib diisi.

**OK**

Password 6 angka  
**085523** 

Outlet Pilih Outlet

---

**Studi Kasus Testing Qasir**

Test Case	Q10
Test Title	Mengisi data Outlet dengan data salah
Description	Memeriksa fitur 'Outlet'
Pre-conditions	Berada di Beranda aplikasi Qasir
Dependencies	Memiliki koneksi internet
Post conditions	Telah masuk dalam fitur 'Tambah Outlet'

No	Test Step	Test Data	Expected result	Actual Result	Status
1	Menekan fitur outlet	Tekan	Menampilkan halaman pecarian	As Expected	Pass
2	Menekan symbol '+'	tekan	Muncul form data outlet	As Expected	Pass
3	Mengisi kolom Nama Cabang	Nama huruf acak	Muncul data yang dimasukkan	As Expected	Fail
4	Mengupload foto outlet	Foto	Muncul foto yang telah diupload	As Expected	Pass
5	Mengisi kolom no. telepon	Nama angka acak	Muncul data yang dimasukkan	Muncul pemberitahuan 'Format no. telpon tidak valid'	Fail
6	Mengisi kolom alamat	Nama huruf acak	Muncul data yang dimasukkan	As Expected	Fail

## Studi Kasus Testing Qasir

No	Test Step	Test Data	Expected result	Actual Result	Status
7	Memilih kelurahan	Nama huruf acak	Muncul data yang dimasukkan	Data tidak muncul dalam pencarian	Fail
8	Mengisi kolom kode pos	Nama angka acak dan kurang dari 6 karakter	Muncul data yang dimasukkan	Muncul pemberitahuan 'Field ini tidak boleh kurang dari 5 dan melebihi 6 karakter'	Fail
9	Mengklik pada lokasi	Klik acak	Data alamat muncul	As Expected	Fail
10	Menekan tombol pilih pegawai	Tekan	Menampilkan pilihan pegawai	As Expected	pass
11	Menekan tombol simpan	Tekan	Muncul pop up keberhasilan	Menampilkan pada form data pegawai	Fail

← **Tambah Outlet** SIMPAN

 Nama Cabang  
ghkulpapakjsbe dhdhd

**Detail Cabang**

No. Telepon  
08746464845469464  
Format No Telepon tidak valid.

Alamat  
gchvjikdnhfssaswsw

Pilih Kelurahan  
Field ini tidak boleh kosong.

Kode Pos  
808  
Field ini tidak boleh kurang dari 5 dan melebihi 6 karakter.

Lokasi

## Kesimpulan dan Saran

Berdasarkan hasil testing yang telah dilakukan, aplikasi Qasir memiliki sangat sedikit bug dan layak digunakan untuk kategori aplikasi tidak berbayar. Pengguna dapat menggunakan semua fitur dengan mudah dan aman. Fitur-fitur yang disediakan meliputi pencatatan transaksi, pengelolaan produk dan stok (penyesuaian, pemindahan, pembelian), pengaturan pegawai dan outlet dan laporan usaha seperti rangkuman penjualan, laporan penjualan harian, laporan penjualan produk. Bahkan untuk memudahkan transaksinya aplikasi Qasir telah mendukung penggunaanya dengan *Bluetooth printer* dan *Bluetooth Barcode Scanner*. Aplikasi Qasir dibuat dengan sangat rapi, detail, cermat dan selalu terupdate untuk memperbaiki bug yang ada sehingga sangat minim bug yang dapat ditemukan pada aplikasi ini. Saran dari kami untuk aplikasi qasir kedepannya adalah dengan menambah opsi untuk menghapus merk dan kategori pada menu kelola produk agar lebih mudah menginputkan barang yang akan dijual.

## Daftar Pustaka

POS Qasir Hadir di 500 Kota Untuk Kembangkan Bisnis UMKM | youngster.id (no date). Available at: <https://youngster.id/news/pos-qasir-hadir-di-500-kota-untuk-kembangkan-bisnis-umkm/> (Accessed: 12 May 2020).

Punya Warung? Dua Aplikasi Ini Patut Dicoba | Kabarjatim.com (no date). Available at: <https://kabarjatim.com/punya-warung-dua-aplikasi-ini-patut-dicoba/> (Accessed: 12 May 2020).

Gimana sih Zaskia Mecca mengatur kesibukan sambil berdagang? | Qasir (no date). Available at: <https://www.qasir.id/studi-kasus/gimana-sih-zaskia-mecca-mengatur-kesibukan-sambil-berdagang> (Accessed: 12 May 2020).

# Studi Kasus Website Cakap.com

Kontributor :

Billah Fathka E.	321810002
Denny Agung S.	321810003
Tansa Wiguna	321810018



## **Latar Belakang**

Cakap.com adalah salah satu perusahaan dalam pembelajaran berbasis online melalui *website*. Sejak tahun 2014 Cakap menjadi pelopor pembelajaran online Di Indonesia. Bahasa Jepang, Mandarin, Inggris, dan bahasa Indonesia adalah pembelajaran bahasa yang disediakan melalui *website* tersebut. Dalam waktu 5 tahun, Cakap.com mengalami pertumbuhan yang cukup pesat yakni 100 ribu lebih pengguna. Berita ini sesuai dengan berita dari *republika.com* (Ramadhan, 2019; Hidayat, 2020)

Cakap.com menarik untuk dikaji adalah karena merupakan salah satu *start up* pembelajaran berbahasa dengan sertifikasi yang kredibel, tenaga kerja profesional, dan kurikulum berstandar nasional dan internasional. Sehingga harus diteliti lebih dalam mengenai proses mendapatkan sertifikasi dan apa saja fitur yang dapat digunakan selama belajar melalui *website* atau aplikasi tersebut. Selain itu, karena menggunakan dasar *website* maka perlu dilakukan testing atau uji coba terhadap segala fungsi yang tersedia. Hal tersebut dilakukan dengan tujuan memastikan setiap fungsi bekerja sesuai dengan tujuan dan standar yang digunakan. Informasi tersebut sesuai dengan laman *website cakap.com* (*Aplikasi Belajar Bahasa Asing Online | Cakap, 2019*)

*Website* yang digunakan harus memiliki standar dengan tujuan menyamakan persepsi setiap karyawan ataupun pelanggan agar terciptanya satu tujuan. Oleh karena itu, fungsi, masukan (input), proses dan keluaran (output) dalam setiap proses harus sesuai dengan kriteria dari standar yang disepakati. Pada tahapan-tahapan proses umumnya memiliki proses yang berbeda-beda. (Ferdiansyah, 2019)

Berdasarkan berita dari [republika.com](http://republika.com), [techno.okezone.com](http://techno.okezone.com), dan informasi dari situs resmi *start up* cakap, aplikasi cakap disediakan dalam bidang jasa pembelajaran *online* dengan berbagai fiturnya. Maka pada bab ini, beberapa fitur utama dengan basis *website* akan dibahas kesesuaiannya.

## **Pembahasan**

Di dalam pembahasan cakap.com ini ada beberapa hal yang ingin ditesting seperti halnya pada tahap registasi, pembayaran dan booking kelas: **registasi**

Registasi terdapat 3 pilihan utama. Menggunakan 3th *party services* (Google-mail dan Facebook) atau mengisi data diri secara manual.

- Password minimal 8-99
- Terdapat confirm pasword
- Semua karakter dapat digunakan
- Memasukan Nomor handphone

	<b>Project Name:</b>		<b>Test Designed by:</b>	<b>Billah Fathka dan Denny Agung</b>
	<b>Module Name:</b>		<b>Test Designed date:</b>	20/04/2020
			<b>Test Executed by:</b>	<b>Tansa Wiguna</b>
			<b>Test Execution date:</b>	21/04/2020 – 22/04/2020

<b>Pre-condition:</b>	User telah berada pada halaman pendaftaran
<b>Dependencies:</b>	User memiliki koneksi internet minimal 50kbps <i>upload</i> dan 100kbps <i>download</i> dengan <i>latency</i> maksimal 100 untuk membuka halaman dengan kecepatan normal

Test Case	Test Title	Test Data	Expected Result	Actual Result	Status	Notes
1	Registrasi dengan email secara manual (bukan 3 <sup>rd</sup> party)	email: +-*/*+*/798@mail.com Nama depan: 1234567-=-`-+*/ Nama belakang: +-*/*`[];'./ Gender: Male Password:	Data yang diberikan invalid atau tidak dapat digunakan	Muncul notifikasi email telah terdaftar pada pendaftaran ke-2	Gagal	- memasukkan segala jenis karakter dapat digunakan untuk melakukan registrasi

**Studi Kasus Website Cakap.com**

		+ - */=_][;'{ }"/.:?!`@#\$\$%^&* ( )				- Persetujuan tanpa <i>checkbox</i> - Tidak ada konfirmasi berhasil mendaftar
2		email: mungkinindia@hotmail.com Nama depan: 1234567-=-`-+*/ Nama belakang: + -*/' [];'./ Gender: Male Password: + - */=_][;'{ }"/.:?!`@#\$\$%^&* ( )	Data yang diberikan invalid atau tidak dapat digunakan	Muncul notifikasi email telah terdaftar pada pendaftaran ke-2	Gagal	- Tidak ada konfirmasi yang masuk mulai tanggal 21/04/2020 hingga 22/04/2020
3		email: mungkinindia1@gmail.com Nama depan: 1234567-=-`-+*/ Nama belakang: + -*/' [];'./ Gender: Male Password:	Data yang diberikan invalid atau tidak dapat digunakan	Muncul notifikasi email telah terdaftar pada pendaftaran ke-2	Gagal	- Tidak ada konfirmasi yang masuk mulai tanggal 21/04/2020 hingga 22/04/2020

		+ */=_][;'{"./,?!`@#\$\$%^&* (				
4		email: sshgoo@gmail.com Nama depan: John Nama belakang: Lenon Gender: Male Password: 123a465b789c	Data yang diberikan valid dan dapat melanjutkan ke halaman berikutnya	Dapat melanjutkan pada halaman berikutnya	Berhasil	- Ditemukan bahwa email valid, <i>password</i> tidak valid tetap akan terpentil dari sistem. Tapi sistem tetap menyimpan email yang telah digunakan untuk <i>login</i>
5	Memilih pelajaran Bahasa	Memilih lebih dari 1 pelajaran	Dapat melakukan <i>multiple-choice</i>	Pilihan tidak dapat dilakukan	gagal	-
6	<i>Input</i> nomor telepon		Tidak dapat lanjut ke halaman berikutnya	Tidak dapat lanjut ke halaman berikutnya dan muncul notifikasi kebutuhan	berhasil	-

## Studi Kasus Website Cakap.com

				nomor telepon		
7		123456789123456	Tidak dapat lanjut ke halaman berikutnya	Tidak dapat lanjut ke halaman berikutnya dan muncul notifikasi nomor terlalu banyak	Berhasil	
8		123456789123	Dapat lanjut ke halaman berikutnya	Dapat lanjut ke halaman berikutnya	Berhasil	<ul style="list-style-type: none"><li>- Perlu fitur keamanan informasi nomor telepon menggunakan verifikasi nomor</li><li>- Perlu fitur keamanan informasi verifikasi email dalam batasan waktu tertentu</li></ul>
9	Login	Email: sshgoo@gmail.com	Dapat melanjutkan	Password invalid	Gagal	

		Password: 123a465b789c	n ke halaman utama			
--	--	---------------------------	--------------------------	--	--	--

*Gambar Registrasi Case 1*

The screenshot shows a registration form titled "by sqline". It contains the following fields and elements:

- Email\***: Input field with "++\*/+-\*/798@mail.com". Below it is a note: "(Submit a valid e-mail to get your progress report)".
- Gender\***: Two radio buttons, "Male" (selected) and "Female".
- First Name\***: Input field with "1234567=-'--+\*/".
- Last Name\***: Input field with "+\*/'[]:/".
- Password\***: Input field with "+\*/=\_][()''./..?!'@#\$\$%\*&'()".
- Confirm Password\***: Input field with "+\*/=\_][()''./..?!'@#\$\$%\*&'()".

Below the form, there is a green "SIGN UP" button. Below the button, the text "Email already taken" is displayed in red.

By signing up you agree to our [Terms And Conditions](#).

*Gambar Registrasi Case2*

The screenshot shows a registration form titled "by sqline". It contains the following fields and elements:

- Email\***: Input field with "mungkindia@hotmail.com". Below it is a note: "(Submit a valid e-mail to get your progress report)".
- Gender\***: Two radio buttons, "Male" (selected) and "Female".
- First Name\***: Input field with "1234567=-'--+\*/".
- Last Name\***: Input field with "+\*/'[]:/".
- Password\***: Input field with "+\*/=\_][()''./..?!'@#\$\$%\*&'()".
- Confirm Password\***: Input field with "+\*/=\_][()''./..?!'@#\$\$%\*&'()".

Below the form, there is a green "SIGN UP" button. Below the button, the text "Email already taken" is displayed in red.

By signing up you agree to our [Terms And Conditions](#).

7

**Gambar Registrasi Case 3**

by squire

Email\*  
mungkindia1@gmail.com  
(Submit a valid e-mail to get your progress report)

Gender\*  
 Male  Female

First Name\*  
1234567-=-'~+\*/

Last Name\*  
+\*/=\_[!()"/./?!'@#%\*&'0

Password\*  
+\*/=\_[!()"/./?!'@#%\*&'0

Confirm Password\*  
+\*/=\_[!()"/./?!'@#%\*&'0

By signing up you agree to our [Terms And Conditions](#).

**SIGN UP**

Email already taken

**Gambar Registrasi Case 4 dan 5**

I Want To Learn...

Mandarin Course

English Course

Japanese Course

English for Kids Course

**NEXT**

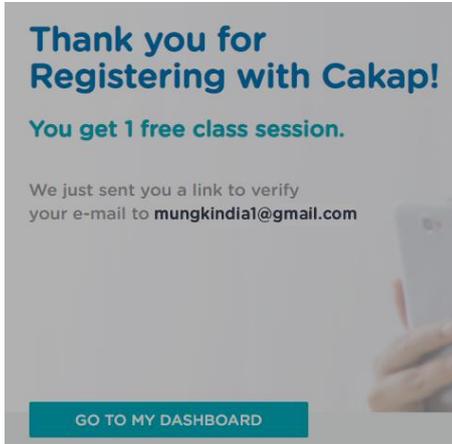
*Gambar Registrasi Case 6*



*Gambar Registrasi Case 7*



*Gambar Registrasi Case 8*



*Gambar Registrasi Case 9*



Email\*

sshgoo@gmail.com

Password\*

123a465b789c

**LOGIN**

Invalid password

The form is part of a login page. It features the Cakap logo (a blue 'C' with a yellow star) and the text "cakap by sqline". There are two input fields: "Email\*" containing "sshgoo@gmail.com" and "Password\*" containing "123a465b789c". A green "LOGIN" button is below the password field. A red error message "Invalid password" is displayed below the button. A small 'x' icon is in the top right corner of the form area, and an eye icon is to the right of the password field.

*Pembayaran*

### ***Studi Kasus Website Cakap.com***

---

Di pembayaran ini pengguna akan disajikan paket-paket yang telah tersedia dengan harga yang relatif tidak terpaut jauh. Informasi paket yang tersedia juga lengkap. Pembayaran dapat melalui bank transfer ataupun *e-wallet*.

Pada bagian *book class* menampilkan 2 pilihan kalender dan *teacher* pada bagian kalender dapat melakukan *booking* jam belajar pada bagian jadwal yang berwarna biru dapat memasukkan jadwal dengan mengisi skype IDE, *timezone*, tanggal lahir dan kota. Di halaman *teacher* dapat melihat guru yang mengajar dan membaca tentang diri guru dengan klik *read more*.

Pada bagian *myclass* dapat menampilkan informasi yang dibutuhkan dan jadwal kelas. Pada bagian *myclass* dapat memilih mata pelajaran yang tersedia dan dapat memilihnya sesuai yang diinginkan. *Buy* kredit menyediakan paket dan harga seta rentang waktu *class* dari setiap opsi pilihan jangka waktu berlangganan. Pada halaman *my invoice* melihat paket yang telah dipilih dan siap dibayar.

	<b>Project Name:</b>		<b>Test Designed by:</b>	Billah Fathka dan Denny Agung
	<b>Module Name:</b>		<b>Test Designed date:</b>	28/04/2020
			<b>Test Executed by:</b>	Tansa Wiguna
			<b>Test Execution date:</b>	28/04/2020 – 29/04/2020

<b>Pre-condition:</b>	<i>User telah berada pada halaman pendaftaran</i>
<b>Dependencies:</b>	<i>User memiliki koneksi internet minimal 50kbps <i>upload</i> dan 100kbps <i>download</i> dengan <i>latency</i> maksimal 100 untuk membuka halaman dengan kecepatan normal</i>

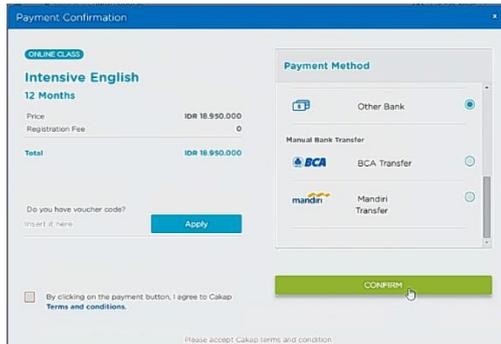
Test Case	Test Title	Test Data	Expected Result	Actual Result	Status	Notes
1	Pembayaran	Tidak melakukan pencentangan <i>check-box payment button. Agreement terms and conditions.</i>	Tidak dapat melanjutkan pembayaran.	Tidak dapat melanjutkan pembayaran.	Berhasil	
2		Melakukan pencentangan <i>check-box payment button. Agreement terms and conditions.</i>	Dapat melanjutkan pembayaran.	Dapat melanjutkan pembayaran.	Berhasil	
3		Mengeklik tombol melihat nomor rekening.	Dapat melihat nomor	Dapat melihat nomor	Berhasil	-

**Studi Kasus Website Cakap.com**

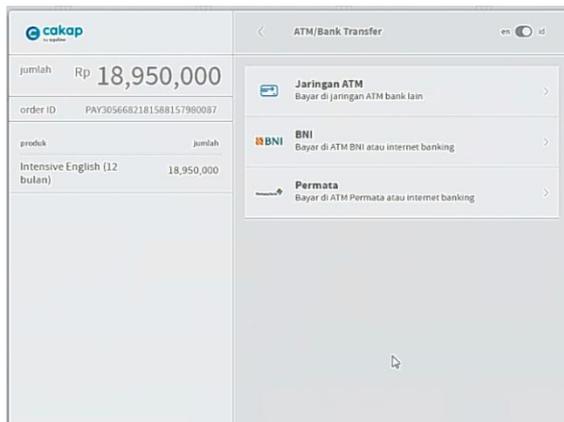
---

			rekening dan besaran biaya.	rekening, besaran biaya, penerbitan invoice, dan panduan pembayaran.		
4	Konfirmasi bukti pembayaran	Transfer Amount: 123456789123 Sender Name 123456789123 Sender Bank Account 123456789123 Receipt DroidCam.Setup.6.0.1.exe	Adanya Batasan Limit pembayaran dan muncul <i>warning</i> file tidak sesuai	Tidak memunculkan notifikasi	Gagal	- Tidak ada <i>warning</i> yang muncul saat melebihi limit pembayaran - Tidak ada <i>warning</i> yang muncul saat <i>file</i> bukti di- <i>upload</i> -

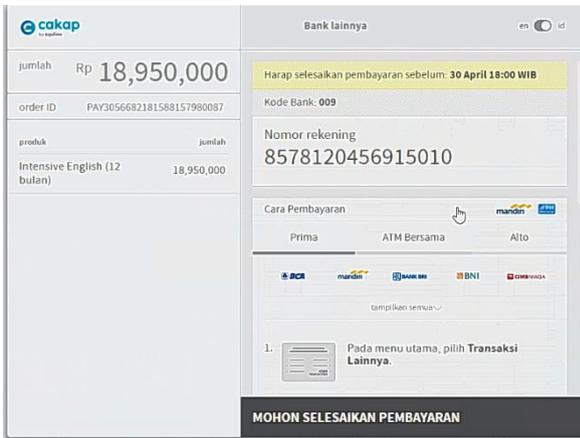
Gambar Pembayaran Case 1



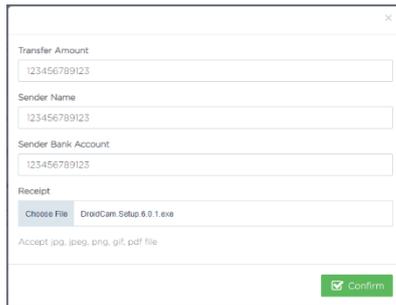
Gambar Pembayaran Case 2



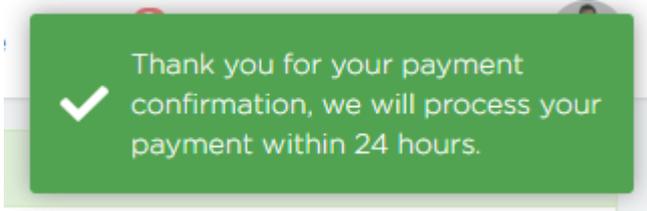
Gambar Pembayaran Case 3



*Gambar Pembayaran Case 4*



*Gambar Pembayaran Case 4*



## Simpulan Dan Saran

*Website* ini masih memiliki celah pada proses bisnis utama dan harus segera diperbaiki. Seperti halnya pada proses registrasi. Ketika pengguna mendaftar dengan *e-mail valid* dan *profile* data sembarang, maka *e-mail* akan tetap dianggap terdaftar. Sehingga pengguna tidak dapat mendaftar lagi dengan *e-mail* tersebut meskipun dengan data yang valid (sesuai data pribadi). Kemudian jika pengguna melakukan *login* dengan *e-mail* dan *password* yang sama dengan registrasi, tetap tidak bisa melakukan *login* untuk menuju halaman *dashboard*. Solusi masalah tersebut adalah dengan mengeklik menu lupa password, dan membuat *password* baru yang sesuai dengan kriteria *password* aplikasi (yang tidak tersedia).

Jadi sebaiknya pada *form* pendaftaran, *developer* mengunci karakter yang tidak sesuai dengan kriteria *login* aplikasi cakap. *Developer* dapat mengunci masukkan pengguna berupa karakter yang tidak umum atau dimaksudkan untuk mengacaukan Bahasa pemrograman. Sehingga pengguna tidak bersusah payah untuk

mencari karakter yang tidak sesuai dengan kriteria karakter pembuatan *password*. Kemudian, *developer* juga membatasi jumlah karakter yang dapat di-inputkan pengguna mulai batas minimum hingga batas maksimum.

Selain registrasi dan *login*, kegiatan pembayaran juga memiliki kelemahan meskipun tidak sampai fatal. Ketika pengguna melakukan pembayaran setelah memilih paket kursus, pengguna harus melakukan *upload* bukti pembayaran disertai data pembayar. Bukti tersebut diolah sistem terlebih dahulu, seperti format bukti berbentuk gambar atau teks atau lainnya. Sehingga validasi hal tersebut dapat dikatakan secara manual oleh karyawan cakap.com. Pengguna dapat melakukan pemalsuan data, yang mana membuat pemeriksaan lebih lambat karena memeriksa bukti palsu yang masuk dalam jajaran transaksi. Belum lagi jika kesalahan validasi dari karyawan cakap.com sendiri.

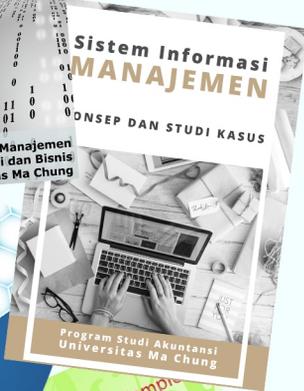
Meskipun menggunakan *virtual account* bank transfer, tetap dibutuhkan bukti fisik pembayaran. Hal ini tetap tabu, karena orang melakukan transfer melalui *virtual account* untuk

memastikan pembayaran berhasil dan ke rekening yang tepat. Sehingga konfirmasi secara langsung antara pihak cakap.com dengan pihak perbankan dapat berlangsung cepat dan tanpa memerlukan bukti lain. Jadi sebaiknya developer tidak mengaktifkan menu konfirmasi pembayaran jika pembayaran secara virtual. Karena hal tersebut berguna mengurangi kepanikan konsumen terhadap pengisian data pembayaran.

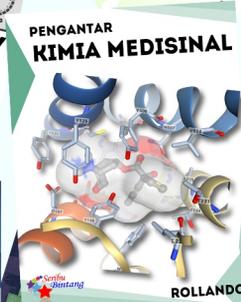
Jadi tingkat keluhan dari cakap.com, berdasarkan hasil testing 2 proses dari kelompok proses bisnis utama. Cakap.com secara keseluruhan berada pada tingkat *serverity* tidak parah. Hal ini didukung sistem keamanan akun yang dapat disinkronkan pada akun *e-mail* dengan ketentuan registrasi menggunakan *3<sup>rd</sup> party account*. Sayangnya meskipun keamanan dari akun yang dapat dikatakan baik (dominan), tapi pembayaran menjadi masalah karena pada saat konfirmasi pembayaran, diperlukan bukti berupa gambar atau foto setruk atau notifikasi bukti pembayaran. Sehingga jika setruk atau notifikasi bukti tersebut hilang, konfirmasi tidak dapat dilakukan dan transaksi gagal.

## Daftar Pustaka

- Aplikasi Belajar Bahasa Asing Online | Cakap* (2019). Available at: <https://cakap.com/> (Accessed: 6 March 2020).
- Ferdiansyah, M. (2019) *Startup Cakap Bantu Anda Belajar Bahasa Asing Secara Online: Okezone techno, techno.okezone.com*. Available at: <https://techno.okezone.com/read/2019/12/16/207/2142649/startup-cakap-bantu-anda-belajar-bahasa-asing-secara-online> (Accessed: 6 March 2020).
- Hidayat, I. W. (2020) *Pengawasan Proses Belajar Online, www.suara.com*. Available at: <https://www.suara.com/yoursay/2020/03/20/094817/pengawasan-proses-belajar-online> (Accessed: 22 April 2020).
- Ramadhan, B. (2019) *Squline Bertransformasi Jadi Cakap | Republika Online, www.republika.co.id*. Available at: <https://www.republika.co.id/berita/trendteknologi/aplikasi/19/04/08/pplop6330-squline-bertransformasi-jadi-cakap> (Accessed: 6 March 2020).



ISBN 978-623-7000-40-2



[www.SeribuBintang.co.id](http://www.SeribuBintang.co.id)  
[info@SeribuBintang.co.id](mailto:info@SeribuBintang.co.id)  
[fb.com/cv.seribu.bintang](https://fb.com/cv.seribu.bintang)  
[seribubintang.gumroad.com](https://seribubintang.gumroad.com)  
IKAPI No. 320/JTI/2021