

**PERANCANGAN PENGUJIAN PERANGKAT LUNAK PADA  
APLIKASI PENDAFTARAN KP DAN TA MENGGUNAKAN  
METODE *BLACK BOX***

**(STUDI KASUS TEKNIK INFORMATIKA UNIVERSITAS PASUNDAN)**

**TUGAS AKHIR**

Disusun sebagai salah satu syarat kelulusan Sidang Tugas Akhir, di  
Program Studi Teknik Informatika, Universitas Pasundan Bandung



Disusun oleh :

Ega Novianingsih

NPM : 17.304.0036

**POGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PASUNDAN BANDUNG**

**JANUARI 2022**

**LEMBAR PENGESAHAN  
LAPORAN TUGAS AKHIR**

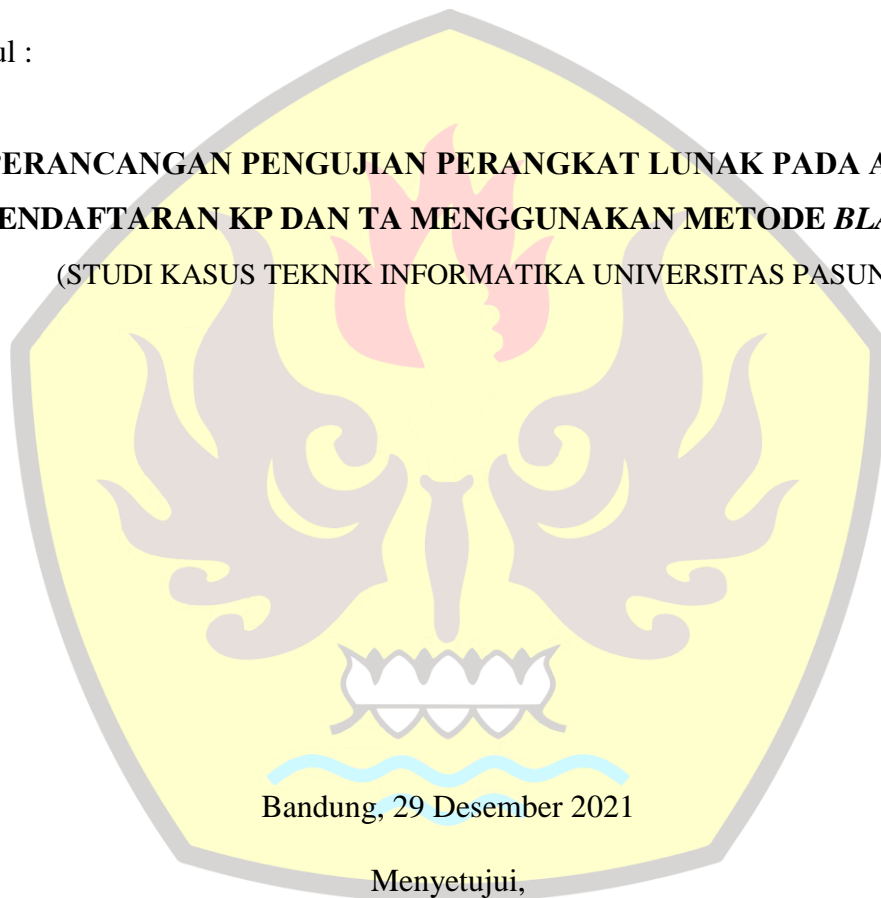
Telah diujikan dan dipertahankan dalam Sidang Sarjana Program Studi Teknik Informatika Universitas Pasundan Bandung, pada hari dan tanggal sidang sesuai berita acara sidang, tugas akhir dari :

Nama : Ega Novianingsih

Nrp : 17.304.0036

Dengan judul :

**“PERANCANGAN PENGUJIAN PERANGKAT LUNAK PADA APLIKASI  
PENDAFTARAN KP DAN TA MENGGUNAKAN METODE *BLACK BOX*  
(STUDI KASUS TEKNIK INFORMATIKA UNIVERSITAS PASUNDAN)”**



Pembimbing Utama,

(Ade Sukendar,S.T,M.T)

Pembimbing Pendamping,

(Wanda Gusdyan,S.T,M.T)

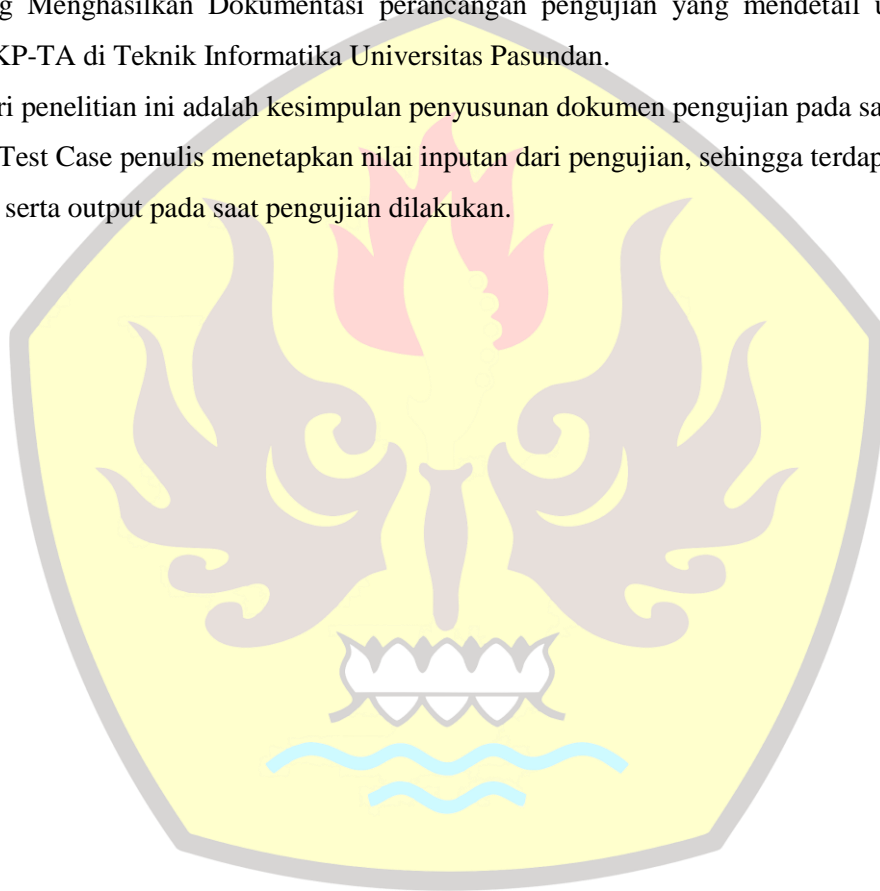


## ABSTRAK

Teknik Informatika Universitas Pasundan merupakan jurusan atau program studi yang menggunakan perangkat lunak berupa rancangan aplikasi Pendaftaran KP-TA untuk membantu dalam proses kegiatan pengelolaan administrasi KP dan tugas akhir mahasiswa. Aplikasi Pendaftaran KP-TA merupakan Aplikasi berbasis website dan mobile. Pengguna Aplikasi adalah Mahasiswa, Dosen Pembimbing, Koordinator KP-TA dan Tata Usaha KP-TA.

Perancangan pengujian Aplikasi pendaftaran KP-TA Teknik Informatika Universitas Pasundan dilakukan dengan metode *Black Box Testing* berbasis *Equivalence Partitioning* dan *Boundary Value Analysis* yang Menghasilkan Dokumentasi perancangan pengujian yang mendetail untuk Aplikasi Pendaftaran KP-TA di Teknik Informatika Universitas Pasundan.

Hasil dari penelitian ini adalah kesimpulan penyusunan dokumen pengujian pada saat melakukan perancangan Test Case penulis menetapkan nilai inputan dari pengujian, sehingga terdapat batasan-batasan input serta output pada saat pengujian dilakukan.



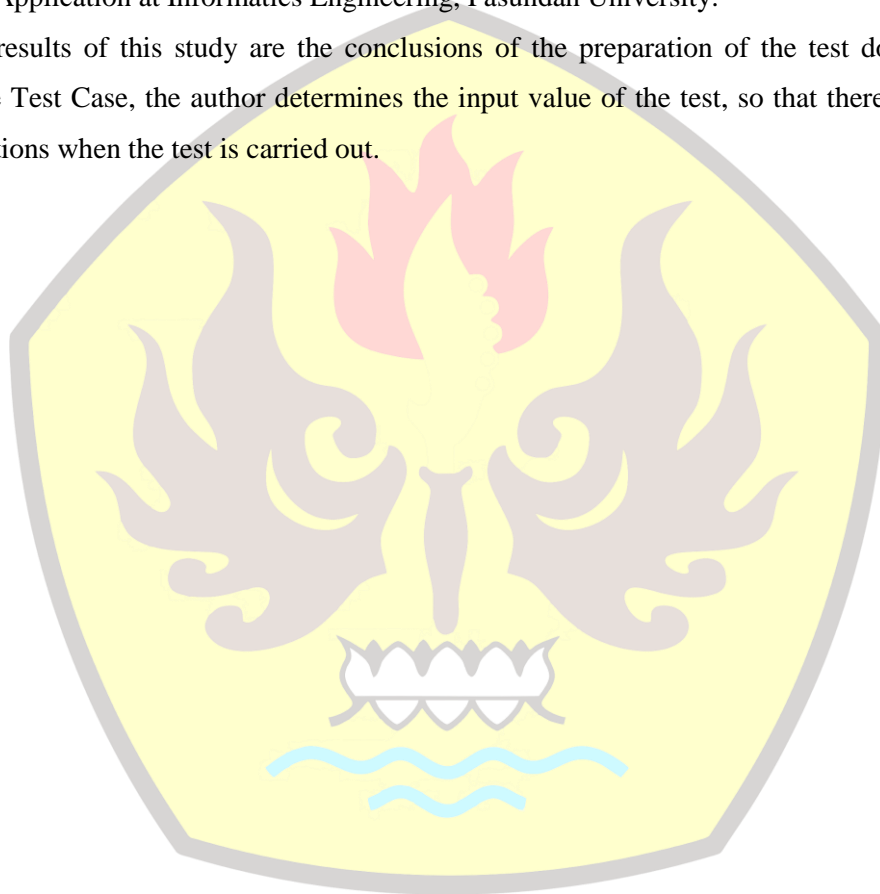


## ABSTRACT

Informatics Engineering at Pasundan University is a department or study program that uses software in the form of a KP-TA Registration application design to assist in the process of managing the KP administration and student final assignments. The KP-TA Registration Application is a website and mobile-based application. Application users are students, Advisory Lecturers, KP-TA Coordinators and KP-TA Administration.

The design of the KP-TA Informatics Engineering registration application at Pasundan University is carried out using the Black Box Testing method based on Equivalence Partitioning and Boundary Value Analysis which produces detailed test design documentation for the KP-TA Registration Application at Informatics Engineering, Pasundan University.

The results of this study are the conclusions of the preparation of the test document when designing the Test Case, the author determines the input value of the test, so that there are input and output limitations when the test is carried out.





## DAFTAR ISI

<b>LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR TABEL.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>BAB 1 PENDAHULUAN.....</b>	<b>1-1</b>
1.1. Latar Belakang .....	1-1
1.2. Identifikasi Masalah .....	1-1
1.3. Tujuan Tugas Akhir .....	1-2
1.4. Lingkup Tugas Akhir .....	1-2
1.5. Metodologi Tugas Akhir .....	1-2
1.6. Skema Penulisan Tugas Akhir .....	1-4
<b>BAB 2 LANDASAN TEORI DAN PENELITIAN TERDAHULU .....</b>	<b>2-1</b>
2.1. Defenisi Pengujian .....	2-1
2.2. Pengujian Perangkat Lunak.....	2-1
2.3. Black Box Testing .....	2-1
2.4. Software Testing Life Cycle(STLC) .....	2-1
2.4.1. Requirement Analysis .....	2-2
2.4.2. Test Planning.....	2-2
2.4.3. Test Case Development.....	2-3
2.4.4. Environment Setup .....	2-3
2.4.5. Test Execution.....	2-3
2.4.6. Test Cycle Closure .....	2-4
2.5. Teknik Pengujian Black Box Testing.....	2-4
2.6. Test Case .....	2-5
2.6.1. Desain Template Test Case .....	2-5
2.7. Functional Testing.....	2-7
2.8. Penelitian Terdahulu .....	2-7
<b>BAB 3 SKEMA PENELITIAN .....</b>	<b>3-1</b>
3.1. Alur Penyelesaian Tugas Akhir.....	3-1
3.2. Perumusan Masalah.....	3-2
3.2.1. Analisis Sebab Akibat .....	3-2
3.2.2. Solusi Masalah .....	3-3
3.3. Kerangka Pemikiran teoritis.....	3-4



3.4. Profile Penelitian .....	3-5
3.4.1. Objek Penelitian .....	3-5
3.4.2. Tempat Penelitian.....	3-5
<b>BAB 4 ANALISIS REQUIREMENT DAN PERANCANGAN TEST CASE.....</b>	<b>4-1</b>
4.1. Analisis Studi Kasus Aplikasi Pendaftaran KP-TA .....	4-1
4.1.1. Pengenalan Aplikasi Pendaftaran KP-TA .....	4-1
4.1.2. Proses Bisnis .....	4-1
4.1.3. User Requirement.....	4-1
4.1.5. Kebutuhan Fungsional.....	4-4
4.2. Test Plan Requirement .....	4-5
4.3. Perancangan Test Case.....	4-6
4.3.1. Equivalence Partitioning .....	4-6
4.3.2. <i>Boundary Value Analysis</i> (BVA) .....	4-10
4.4. Daftar Rancangan Test Case .....	4-13
4.5. Test Case .....	4-13
4.6. Hasil Review Test Case.....	4-21
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>	<b>5-1</b>
5.1. Kesimpulan .....	5-1
5.2. Saran.....	5-1
<b>DAFTAR PUSTAKA .....</b>	<b>5-2</b>
<b>LAMPIRAN.....</b>	<b>5-4</b>





## **BAB 2**

### **LANDASAN TEORI DAN PENELITIAN TERDAHULU**

#### **2.1. Defenisi Pengujian**

Definisi pengujian perangkat lunak adalah proses untuk mencari kesalahan pada setiap item perangkat lunak, mencatat hasilnya, mengevaluasi setiap aspek pada setiap komponen sistem dan mengevaluasi semua fasilitas dari perangkat lunak yang dikembangkan. Terdapat 2 hal utama yang dilakukan dalam pengujian, yaitu [AHI20].

1. Verifikasi adalah proses mengevaluasi suatu system/component untuk menentukan apakah suatu produk yang diselesaikan setelah fase pengembangan memenuhi kondisi seperti yang telah ditetapkan pada awal pengembangan (saat menentukan spesifikasi) perangkat lunak. (*“Are we building the product right?”*) [AHI20].
2. Validasi adalah proses mengevaluasi suatu sistem atau komponen pada akhir atau selama masa pengembangan untuk menentukan apakah produk yang dihasilkan telah memenuhi kebutuhan-kebutuhan dan persyaratan tertentu yang diminta oleh *user*. (*“Are we building the right product?”*) [AHI20].

Dalam melakukan pengujian harus memilih teknik yang tepat, yaitu teknik yang dapat menemukan kesalahan yang belum terdeteksi sehingga dapat meningkatkan kualitas software [YUL20].

#### **2.2. Pengujian Perangkat Lunak**

Pengujian Perangkat lunak merupakan proses eksekusi suatu program atau sistem dengan maksud menemukan atau, melibatkan setiap kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan suatu program atau sistem dan menentukan bahwa itu memenuhi hasil yang dibutuhkan perusahaan [KOM16].

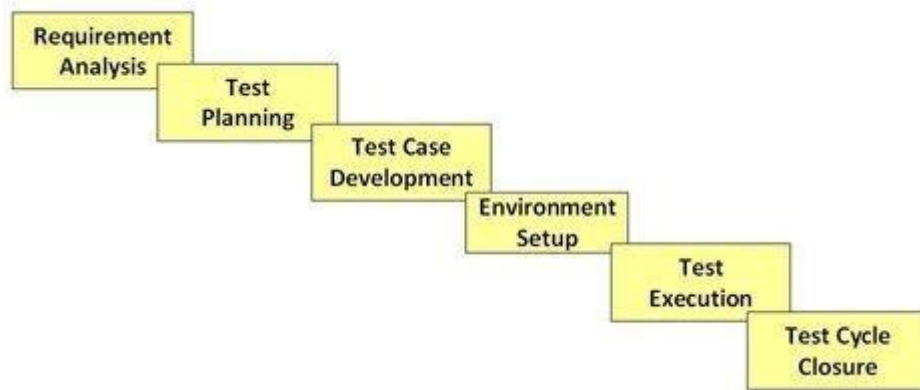
#### **2.3. Black Box Testing**

*Black-Box Testing* merupakan pengujian berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program [KOM16]. Pengujian mengartikan sebuah kondisi masukan seta melakukan sebuah pengujian bagi perincian peranan software [RIZ20].

#### **2.4. Software Testing Life Cycle (STLC)**

*Software Testing Life Cycle* adalah urutan aktivitas spesifik yang dilakukan selama proses pengujian untuk memastikan tujuan kualitas perangkat lunak terpenuhi.





**Gambar 2-1 Software Testing Life Cycle(STLC)**

#### **2.4.1. Requirement Analysis**

Analisis Kebutuhan adalah langkah yang paling pertama dalam *Software Testing Life Cycle* (STLC). Dalam langkah ini, tester mempelajari persyaratan dari sudut pandang pengujian untuk mengidentifikasi persyaratan yang dapat diuji. Tim QA dapat berinteraksi dengan berbagai Stakeholder (*Client, Business Analyst, Technical Leads, System Architects*, dll) untuk memahami persyaratan secara rinci [GUR16].

Persyaratan dapat berupa Fungsional (mendefinisikan apa yang harus dilakukan perangkat lunak) atau Non Fungsional (mendefinisikan kinerja sistem / ketersediaan keamanan). Kegiatan:

1. Buat daftar untuk Jenis Pengujian seperti Fungsional, Keamanan, Kinerja dll.
2. Tentukan fokus dan prioritas pengujian.
3. Persiapkan Requirement Traceability Matrix (RTM).
4. Buat daftar rincian lingkungan Uji di mana kegiatan pengujian akan dilakukan.
5. Periksa kelayakan Otomasi jika diperlukan & siapkan laporan kelayakan Otomatisasi.

#### **2.4.2. Test Planning**

*Test Planning* adalah fase paling penting dari siklus hidup pengujian Perangkat Lunak di mana semua strategi pengujian didefinisikan. Fase ini juga disebut sebagai tahap Uji Strategi. Pada fase ini biasanya Test Manager (atau Test Lead berdasarkan perusahaan ke perusahaan) terlibat untuk menentukan upaya dan perkiraan biaya untuk keseluruhan proyek. Fase ini akan dimulai setelah fase pengumpulan persyaratan diselesaikan & berdasarkan analisis kebutuhan, mulai mempersiapkan Test Strategy. Hasil dari tahap Test Planning adalah Test Plan atau Test Strategy & Pengujian dokumen estimasi usaha. Setelah tahap Test Planning selesai, tim QA dapat memulai dengan kegiatan pengembangan kasus uji [GUR16] Kegiatan:

1. Tentukan Tujuan & ruang lingkup proyek.
2. Buat daftar jenis pengujian yang terlibat dalam STLC.
3. Estimasi uji coba dan perencanaan sumber daya. Pemilihan alat pengujian jika diperlukan.
4. Tentukan ikhtisar proses pengujian.

5. Tentukan lingkungan pengujian yang diperlukan untuk seluruh proyek.
6. Siapkan jadwal tes. Tentukan prosedur kontrol.
7. Menentukan peran dan tanggung jawab. Buat daftar pengiriman pengujian.
8. Tentukan kriteria masuk, kriteria suspensi, kriteria kembalinya dan kriteria keluar.
9. Tentukan risiko yang terlibat jika ada.

#### **2.4.3. Test Case Development**

Pembangunan *Test Case* dilakukan setelah aktifitas *Test Planning* telah selesai. Fase ini melibatkan pembuatan, verifikasi dan pengerjaan ulang kasus uji & skrip pengujian. Data uji, diidentifikasi / dibuat dan ditinjau dan juga kemudian dikerjakan ulang [GUR16] Kegiatan:

1. Buat uji kasus, skrip otomatisasi (jika ada)
2. Tinjau dan uji dasar kasus dan skrip
3. Buat data uji (Jika Lingkungan Uji tersedia)

#### **2.4.4. Environment Setup**

Pada dasarnya lingkungan uji memutuskan pada kondisi mana perangkat lunak diuji. Ini adalah aktivitas independen dan dapat dimulai paralel dengan *Test Case Development*. Dalam proses pengaturan tim uji lingkungan pengujian tidak terlibat di dalamnya. Berdasarkan perusahaan ke perusahaan mungkin pengembang atau pelanggan menciptakan lingkungan pengujian. Berarti sementara tim penguji harus menyiapkan kasus uji asap untuk memeriksa kesiapan pengaturan lingkungan pengujian [GUR16] Kegiatan:

1. Analisis persyaratan dan siapkan daftar Perangkat Lunak & perangkat keras yang diperlukan untuk mengatur lingkungan pengujian.
2. Pengaturan test Environment dan test data
3. Setelah Test Environment diset untuk mengeksekusi smoke test case untuk memeriksa kesiapan lingkungan pengujian.

#### **2.4.5. Test Execution**

Selama fase ini penguji akan melakukan pengujian berdasarkan pada rencana uji dan kasus uji yang disiapkan. Bug akan dilaporkan kembali ke tim pengembangan untuk koreksi dan pengujian ulang akan dilakukan [GUR16] Kegiatan:

1. Jalankan tes sesuai rencana
2. Tandai status kasus uji coba seperti Passed, Failed, Blocked, Not Run, dll.
3. Mendokumentasikan hasil pengujian, dan mencatat kerusakan untuk kasus yang gagal
4. Tetapkan ID Bug untuk semua kasus uji yang Gagal dan Diblokir.
5. Lakukan Pengujian ulang setelah cacat diperbaiki.

#### 2.4.6. Test Cycle Closure

Rapat anggota tim penguji & evaluasi kriteria penyelesaian siklus berdasarkan cakupan Uji, Kualitas, Biaya, Waktu, Sasaran Bisnis Kritis, dan Perangkat Lunak. Diskusikan semua yang berjalan baik, area mana yang perlu ditingkatkan & mengambil pelajaran dari STLC saat ini sebagai masukan untuk siklus pengujian yang akan datang, yang akan membantu meningkatkan hambatan dalam proses STLC. Test case & bug report akan menganalisa untuk mengetahui distribusi cacat berdasarkan jenis dan tingkat keparahannya. Setelah menyelesaikan siklus uji maka laporan penutupan tes & Metrik uji akan disiapkan. Analisis hasil pengujian untuk mengetahui distribusi cacat berdasarkan jenis dan tingkat keparahannya [GUR16] Kegiatan:

1. Evaluasi kriteria penyelesaian siklus berdasarkan cakupan Uji, Kualitas, Biaya, Waktu, Tujuan Bisnis Kritis, dan Perangkat Lunak.
2. Siapkan metrik tes berdasarkan parameter di atas.
3. Siapkan laporan penutupan Uji. Analisis hasil pengujian untuk mengetahui distribusi cacat berdasarkan jenis dan tingkat keparahannya.

#### 2.5. Teknik Pengujian Black Box Testing

Teknik pengujian *black box* merupakan teknik yang digunakan dalam melakukan pengujian menggunakan metode *black box*.

##### 1. Equivalence Partitioning (Partisi ekuivalensi)

*Equivalence Partitioning* mendefinisikan kasus yang akan diuji dengan menemukan beberapa kesalahan dan meminimalisir jumlah kasus yang harus dibuat. Design test case equivalence class partitioning berdasarkan pada input dan output dari suatu komponen yang dipartisi ke dalam kelas-kelas sesuai dengan spesifikasi dari komponen itu, yang diperlakukan sama (equivalence). Bisa juga dianggap bahwa inputan akan menghasilkan respon yang sama [KIK19].

*Equivalence Partitioning* adalah strategi yang dapat digunakan untuk mengurangi jumlah test case yang akan dibuat dikarenakan tujuan ujinya sama. Penyetaraan golongan dengan cara membagi nilai input menjadi beberapa kondisi. Contohnya, input data mengenai nilai. Jika nilai dibawah 30 maka tidak bisa di input, pembagian kondisinya sebagai berikut

**Table 2-1 Equivalence Partitioning (Partisi ekuivalensi)**

Valid	Invalid
Nilai $\geq 30$	Nilai $< 30$

Setelah membuat pembagian kondisinya maka selanjutnya membuat test case dari masing-masing pembagian kondisi. Test case di kelompokkan menjadi 2, yaitu nilai valid dan tidak valid.

## 2. *Boundary Value Analysis* (Analisis Nilai Batas)

Dalam pengujian *Boundary Value Analysis* dilakukan melalui tahapan setiap modul dan menu yang ada pada Sistem Informasi Pendaftaran dan Pembayaran terintegrasi Wisuda Online, pengujian dengan *Boundary Value* sehingga akan didapat nilai prosentase dari kegunaan system informasi ini[AHI20]. *Boundary Value Analysis* lebih mencermati kepada jumlah angka dan huruf yang dimasukkan dan hasil yang dikeluarkan dari data yang dimasukkan tersebut dan penyesuaian nilai dengan nilai maksimum dan nilai minimum[SOE20].

### 2.6. Test Case

*Test case* adalah seperangkat instruksi pada "CARA" untuk memvalidasi tujuan / sasaran uji tertentu, yang ketika diikuti akan memberi tahu kita apakah perilaku yang diharapkan dari sistem terpenuhi atau tidak. Sebuah test case memiliki komponen yang menggambarkan input, aksi / kejadian dan respon yang diharapkan, untuk menentukan apakah fitur aplikasi berfungsi dengan benar.

#### 2.6.1. Desain Template Test Case

Teknik desain Test Case yang baik sangat penting untuk meningkatkan kualitas proses pengujian perangkat lunak. Ini membantu meningkatkan kualitas dan efektivitas keseluruhan perangkat lunak yang dirilis. Berikut ini merupakan tabel template test case yang dipakai untuk menerapkan uji skenario pada setiap fungsionalitas perangkat lunak.[KIK19].

**Table 2-2 Desain Template Test Case**

<b>Identifikasi(1)</b>	No/ID Test Case
<b>Nama Uji Kasus (2)</b>	Nama Test Case
<b>Deskripsi (3)</b>	Deskripsi test case
<b>Kondisi Awal (4)</b>	Perangkat lunak menampilkan halaman dashboard
<b>Test Data (5)</b>	View data
<b>Tanggal Pengujian (6)</b>	
<b>Penguji (7)</b>	
<b>Skenario (8)</b>	
Kondisi : (9)	



1.		
2.		
<b>Hasil (10)</b>		
<b>Yang Diharapkan (11)</b>	<b>Kesimpulan (12)</b>	
	<b>PAS S</b>	<b>FAI L</b>
1. Menampilkan list catatan orang tua atau wali murid	√	
2. Menampilkan data berhasil disimpan	√	

Deskripsi:

1. **Identifikasi**, merupakan identifikasi nomor/id test case diisi dengan “TC” artinya test case dan di akhiri dengan nomor kebutuhan fungsionalitas perangkat lunak “01”
2. **Nama Kasus Uji**, diisi dengan nama kasus uji atau nama test case berdasarkan nama yang telah dibuat pada saat pembuatan test plan
3. **Deskripsi**, merupakan penjelasan tentang gambaran mengenai test case tersebut
4. **Kondisi awal**, merupakan keadaan awal sebelum skenario awal di eksekusi
5. **Test Data**, merupakan input yang digunakan dalam skenario test case tersebut
6. **Tanggal pengujian**, merupakan tanggal pengujian dilaksanakan
7. **Penguji**, merupakan nama penguji yang menguji perangkat lunak tersebut 2-10
8. **Skenario**, langkah-langkah yang harus dilakukan berurutan untuk dapat menguji kesesuaian hasil yang diharapkan dari pengujian tersebut.
9. **Kondisi**, merupakan identifikasi kondisi yang ingin diuji. Misalnya “menginput dengan nilai yang valid”
10. **Hasil**, merupakan kolom untuk keterangan dan pengisian dari hasil yang diharapkan dan kesimpulan dari skenario yang disediakan pada kolom skenario
11. **Yang diharapkan**, hasil yang diharapkan keluar ketika tahapan skenario telah dilakukan dengan benar
12. **Kesimpulan**, merupakan keterangan dari “PASS” dan “FAIL” dari hasil pengujian test case. Keterangan “PASS” akan diberi tanda checklist apabila sesuai dengan hasil yang diharapkan,

dan keterangan “FAIL” akan diberi tanda checklist apabila hasil yang diharapkan tidak sesuai.



## 2.7. Functional Testing

Pengujian fungsional adalah jenis pengujian perangkat lunak yang memvalidasi sistem perangkat lunak terhadap persyaratan / spesifikasi fungsional. Tujuan pengujian Fungsional adalah untuk menguji setiap fungsi aplikasi perangkat lunak, dengan memberikan masukan yang sesuai, memverifikasi keluaran terhadap persyaratan Fungsional.[SPA17]

## 2.8. Penelitian Terdahulu

Penelitian terdahulu disampaikan sebagai rujukan utama dalam tugas akhir. Penelitian terdahulu ditetapkan karena kita menggunakan beberapa pemikiran dari penelitian lain, memanfaatkan produk penelitiannya, atau sebagai pembanding keberhasilan tugas akhir kita.

Berikut adalah daftar penelitian-penelitian terdahulu yang mendukung dan memiliki kemiripan dengan tugas akhir yang berjudul Pengujian Perangkat Lunak Menggunakan Metode Black Box (Studi Kasus Aplikasi KP-TA Fakultas Teknik Universitas Pasundan), sebagaimana dijelaskan pada table 2.1.

**Table 2-3 Ringkasan Penelitian Terdahulu**

No	Judul Penelitian	Peneliti	Hasil	Metode yang Digunakan
1	Pengujian Perangkat Lunak pada Aplikasi KP-TA	Kiki Hadiansyah[KIK19]	Menguji perangkat lunak pada aplikasi KP-TA	Penggunaan Metode <i>Equivalence Partitioning</i> dan Penggunaan <i>Boundary Value Analysis</i>
2	Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan <i>Black Box Testing</i> dengan Metode <i>Equivalence Partitioning</i> dan <i>Boundary Value Analysis</i> universitas Islam Syekh Yusuf	Taufik Hidayat[HID18]	pengujian yang efektivitas dan fungsionalitas dari suatu system informasi pendaftaran dan pembayaran wisuda online di UNIS	Penggunaan Metode <i>Equivalence Partitioning</i> dan Penggunaan <i>Boundary Value Analysis</i>

3	Pengujian Perangkat Lunak Menggunakan Metode Black Box Studi Kasus Exelsa Universitas Sanata Dharma	Ganang Wahyu Setiawan[WAH 11]	Memberikan dokumentasi hasil pengujian yang menginformasikan kesesuaian perangkat lunak yang diuji dengan spesifikasi yang telah ditentukan dan menemukan kesalahan-kesalahan yang terdapat pada EXELSA	Penggunaan Metode Black Box Testing
---	-----------------------------------------------------------------------------------------------------	-------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------



4	Pengujian Perangkat Lunak Metode Black Box berbasis Equivalence partitions pada aplikasi system informasi sekolah	Komarudin MZ[KOM16]	menunjukkan bahwa terdapat celah eror pada beberapa form perangkat lunak system informasi sekolah yang ada di SDIT Al Qudwah Punggur Lampung Tengah.	Penggunaan Metode Black Box berbasis Equivalence partitions
5	Pengujian perangkat lunak dengan menggunakan model behavior UML	Waskitho Wibisono[WIB02]	Menguji perangkat lunak dengan menggunakan model behavior UML	Penggunaan dengan model Bahavoir UML

Penjelasan penelitian terdahulu sebagai berikut :

1. Literatur pertama, membahas tentang perangkat lunak
2. Literatur kedua, membahas tentang konsep pengujian menggunakan metode black box
3. Literatur ketiga, membahas tentang konsep pengujian dengan basis Equivalence partitions dan *Boundary Value Analysis*
4. Literature keempat, membahas tentang skema analisis
5. Literature kelima, membahas tentang konsep perancangan model pengujian perangkat lunak

## DAFTAR PUSTAKA

- [AHI20] Ahr Ahizal, D., Khaerul Miftah, M., Kurniawan, R., & Zaelani, T. (2020). Pengujian Perangkat Lunak Sistem Informasi Peminjaman PlayStation dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. 5(1).  
<http://openjournal.unpam.ac.id/index.php/informatika73>
- [GUR16] Guru99 Team. (2016). STLC - Software Testing Life Cycle.  
<https://www.guru99.com/software-testing-life-cycle.html> ,(08 Februari 2021)
- [HID18] Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan *Black Box Testing* dengan Metode *Equivalence Partitioning* dan *Boundary Value Analysis*. In Jurnal Teknik Informatika UNIS JUTIS (Vol. 6, Issue 1). [www.ccsenet.org/cis](http://www.ccsenet.org/cis)
- [KIK19] Kiki Hadiansyah. (2019). PENGUJIAN PERANGKAT LUNAK PADA APLIKASI PEMANTAUAN KEGIATAN SISWA (Studi Kasus: Madrasah Ibtidaiyah Cahaya Cimahi).
- [KOR16] Komarudin, M. (2016). PENGUJIAN PERANGKAT LUNAK METODE BLACK-BOX BERBASIS EQUIVALENCE PARTITIONS PADA APLIKASI SISTEM INFORMASI SEKOLAH.
- [WAH11] Wahyu Setiawan, G. (2011). SOFTWARE TESTING USING BLACK BOX METHOD ON SANATA DHARMA UNIVERSITY'S EXELSA.
- [WIB02] Wibisono, W., & Baskoro, F. (2002). PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN MODEL BEHAVIOUR UML.
- [SPA17] Spaceku Team., "Pengertian Functional Testing". Diambil dari:  
<https://spaceku.com/pengertian-functional-testing/> , 2017 (07 Januari 2021)
- [GUR20] Guru99 (2020). Boundary Value Analysis dan *Equivalence Partitioning* Testing.  
<https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html> ,(12 november 2021)
- [MAD19] Made I, Ardana S. PENGUJIAN SOFTWARE MENGGUNAKAN METODE BOUNDARY VALUE ANALYSIS DAN DECISION TABLE TESTING. Vol 40.; 2019.
- [PUT20] Putra Pratama B, Bagus Vitriadi Ristianto I, Adi Prayogo I, Saifudin A. Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa Dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. Vol 1.; 2020.  
<http://openjournal.unpam.ac.id/index.php/JOAIIA32>
- [YUL20] Yulistina SR, Nurmala T, Supriawan R, Saifudin A. Penerapan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Penjualan Menggunakan Metode Black Box Testing. Jurnal Informatika Universitas Pamulang. 2020;5(2):129.  
doi:10.32493/informatika.v5i2.5366

- [RIZ20] Rizaldi MI, Ramadhan S, Majid MNH, Yulianti Y. Pengujian Black Box pada Sistem Informasi Praktek Kerja Industri (PRAKERIN) dengan menggunakan Metode Boundary Value Analysis. Jurnal Teknologi Sistem Informasi dan Aplikasi. 2020;3(3):165. doi:10.32493/jtsi.v3i3.5318
- [SOE20] Soejono DDW, Candra DR, Rosandi D, Putra RR, Nurfaidzin Y, Ardiansyah H. Pengujian Black Box pada Aplikasi Penjualan Berbasis Website dengan PHP MySQL Kedai Kopi.Net Menggunakan Teknik Boundary Value Analysis. Jurnal Teknologi Sistem Informasi dan Aplikasi. 2020;3(2):90. doi:10.32493/jtsi.v3i2.3864

