PENGUJIAN SOFTWARE MENGGUNAKAN METODE BOUNDARY VALUE ANALYSIS DAN DECISION TABLE TESTING

I Made Sugi Ardana

Program Studi Teknik Informatika STMIK Eresha

Jl. Raya Puspiptek No.11, Buaran, Serpong, Kota Tangerang Selatan, Banten 15310

E-mail : sugiardana@gmail.com

ABSTRAK

Kualitas produk sangat penting dalam industri pembuatan software. Kualitas produk yang dihasilkan sangat dipengaruhi oleh pengujian software sebelum suatu software dideploy ke pelanggan. Jika suatu software telah lulus uji, maka akan dapat diminimalisir terjadinya kendala pada saat software dipergunakan atau dioperasikan. Terdapat banyak metode pengujian software yang dapat digunakan, dari jenis Black Box Testing maupun jenis White Box Testing. Pada penelitian ini digunakan metode dari jenis Black Box Testing, yaitu Boundary Value Analysis dan Decision Table Testing. Sampel form yang diuji merupakan bagian dari software yang dipergunakan untuk pengelolaan Dana Pensiun untuk suatu lembaga pegelolaan dana pensiun karyawan. Sampel form yang diuji adalah form Pengajuan Klaim Manfaat Pensiun Berkala. Lalu dilakukan pengujian dengan menggunakan metode Boundary Value Analysis dan Decision Table Testing. Berdasarkan aturan proses bisnis entri dan validasi form, dibuat skenario pengujian dan data yang akan digunakan untuk menguji. Setelah melalui proses pengujian diperoleh hasil bahwa masih ada yang harus diperbaiki pada sampel form tersebut.

***Kata Kunci** : Pengujian; Software; Boundary Value Analysis; Decision Table Testing; STLC*

1. PENDAHULUAN

Kegunaan utama dari pengujian software adalah untuk mendeteksi kesalahan yang terdapat pada software sehingga kesalahan tersebut dapat diungkap dan diperbaiki [1].

Pembuatan suatu software mengacu pada suatu kebutuhan yang telah didefinisikan dan disepakati sebelumnya. Pengujian software diperlukan untuk melakukan verifikasi dan validasi bahwa software yang telah dibangun sudah sesuai dengan kebutuhan tersebut [2]. Jika software yang dibangun tidak sesuai dengan kebutuhan yang telah didefinisikan sebelumnya, maka klien akan kecewa dan perusahaan pembuat software tersebut akan kehilangan potensi klien.

Jika klien merasa puas dengan software yang dibuat oleh suatu perusahaan pembuat software, biasanya akan bercerita ke relasinya. Maka hal ini akan membuka peluang dan potensi adanya klien baru yang akan memakai software yang dibangun. Maka pengujian software menjadi sangat penting untuk keberlangsungan usaha suatu perusahaan pembuat software.

Pada penelitian ini dilakukan studi di perusahaan pembuat software yang membuat software berdasarkan kebutuhan klien yang spesifik. Perusahaan ini membuat software untuk operasional perusahaan asuransi, untuk pengelolaan keuangan perusahaan, dan untuk operasional perusahaan pengelolaan dana pensiun. Dari beberapa jenis software yang dibuat tersebut, pada penelitian ini diambil sampel software yang digunakan untuk pengelolaan dana pensiun suatu perusahaan. Lalu dari software tersebut diambil sampel form Pengajuan Klaim Manfaat Pensiun Berkala.

Terdapat banyak metode yang dapat digunakan untuk melakukan pengujian software. Metode *Boundary Value Analysis* merupakan salah metode mendeteksi kesalahan dengan membatasi input pada batasan tertentu, suatu nilai di atas atau di bawah suatu batasan [3]. Pada penelitian ini, metode ini digunakan untuk pengujian batasan input Pengajuan Klaim Manfaat Pensiun Berkala.

Metode *Decision Table Testing* menggunakan kombinasi beberapa kondisi untuk mengambil keputusan [3]. Pada penelitian ini, metode ini digunakan untuk menentukan kombinasi input mana yang akan memicu perhitungan manfaat pensiun berkala.

Pada penelitian ini metode *Boundary Value Analysis (BVA)* dan *Decision Table Testing* dipergunakan untuk melakukan pengujian software untuk validasi input yang berbeda namun saling melengkapi fungsi form.

2. TINJAUAN PUSTAKA

2.1 Pengujian Software

Pengujian software penting karena setiap orang membuat kesalahan pada saat pembuatan software. Maka akan terjadi kesalahan, defect, dan bug yang berbeda pada software. Beberapa defect mungkin

menyebabkan hasil yang fatal, dan beberapa mungkin tidak. Hal ini tergantung dari level dari defect tersebut. Seorang tester harus melakukan pengecekan untuk menghasilkan software yang memuaskan para pemakai. Tidak ada software yang 100% tanpa *defect* [3].

Pengujian software sangat diperlukan untuk memastikan software/aplikasi yang sudah/sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang atau penguji software harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean [4].

Pengujian software mengikuti beberapa prinsip untuk membantu proses pengembangan software. Prinsip pengujian meliputi: [3]

1. Pengujian menunjukkan keberadaan *defect*. Pengujian adalah proses untuk menemukan defect. Tidak ada software yang 100% tanpa *defect*. Setiap software memiliki *defect*. Ada defect yang berbahaya dan ada yang tidak. Pengujian membantu menemukan *defect* tersebut sehingga bisa dihilangkan.

2. Pengujian secara lengkap tidak mungkin dilakukan. Untuk menguji setiap elemen atau setiap kombinasi pada software tidaklah mungkin. Penguji memilih hanya bagian yang terkait dengan suatu skenario pengujian.

3. Lakukan pengujian lebih awal. Semakin cepat dilakukan pengujian semakin baik kita dapat menggunakan waktu. Pengujian harus dilakukan pada setiap fase pengembangan software.

4. Pesticide Paradox. Jika skenario pengujian yang sama dilakukan secara berulang-ulang maka tidak akan ditemukan defect baru. Maka untuk melakukan pengujian harus dibuat beberapa skenario pengujian yang berbeda-beda.

5. Defect Clustering. Selama proses pengujian, akan ditemukan bahwa sebagian besar defect terkait dengan sebagian kecil modul software.

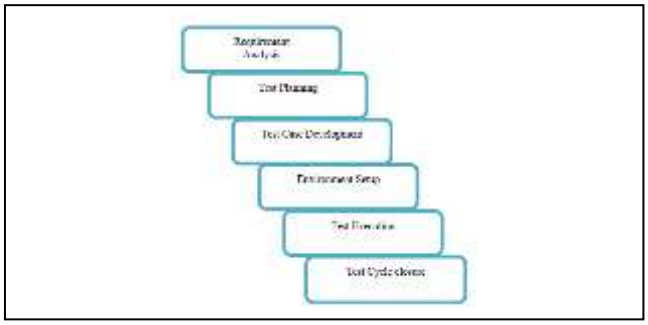
6. Pengajuan tergatung konteks. Teknik pengujian yang berbeda dilakukan sesuai dengan jenis sofware yang diuji. Sebagai contoh, untuk menguji software kesehatan akan berbeda tekniknya dengan software game.

7. Tidak adanya kesalahan adalah sebuah kesesatan. Tidak adanya defect tidak menunjukkan bahwa software akan menghasilkan hasil yang akurat. Kadang-kadang hasil yang buruk dihasilkan setelah software diserahterimakan. Maka penting melakukan pengujian setelah serah terima software.

2.2 STLC (Software Testing Life Cycle)

STLC merupakan siklus proses pengujian yang dijalankan secara sistematis dan dengan perencanaan yang baik. Pada STLC beberapa aktivitas berbeda dilakukan untuk meningkatkan kualitas produk. STLC merupakan panduan bagi para pelaksana dan manajemen sehingga perkembangannya dapat terukur dalam upaya mencapai suatu milestone [1].

Seperti ditunjukkan pada Gbr. 1 terdapat beberapa tahapan siklus, dimulai dari *Requirement Analysis* untuk menganalisis kebutuhan, *Test Planning* untuk pembuatan rencana pengujian, *Test Case Development* untuk membuat skenario pengujian, *Environment Setup* untuk menyiapkan sarana pendukung pengujian, *Test Execution* untuk melaksanakan pengujian dan *Test Cycle Closure* untuk membuat kesimpulan hasil pengujian.



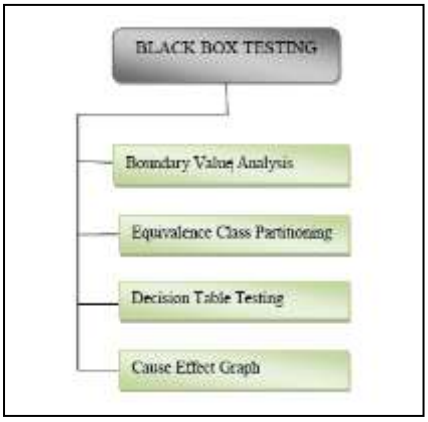
Gbr. 1 STLC (*Software Testing Life Cycle*). [1]

2.3 Black Box Testing

Black box testing juga disebut pengujian berdasarkan fungsional atau spesifikasi. Black box ini melibatkan pengamatan terhadap output berdasarkan input tertentu. Tidak mempertimbangkan detail software. Hanya diperiksa nilai output berdasarkan nilai input masing-masing. Tidak ada upaya untuk mempelajari atau memeriksa kode program bagaimana output diproduksi. Pengujian ini didasarkan pada spesifikasi eksternal. Hanya memeriksa fungsionalitas perangkat lunak, mengamati aspek-aspek dasar dari software, untuk memeriksanya apakah itu sesuai dengan kebutuhan pengguna [3].

Pada black box testing penguji dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Black box testing bukanlah solusi alternatif dari White box testing, tetapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh White box testing [4].

Terdapat beberapa strategi yang dapat digunakan untuk melakukan Black box testing, diantaranya *Boundary Value Analysis*, *Equivalence Class Partitioning*, *Decision Table Testing*, *Cause Effect Graph* seperti ditunjukkan pada Gbr.2.



Gbr. 2 Strategi Black Box Testing. [3]

Black box testing memiliki beberapa kelebihan jika dibandingkan dengan white box testing, yaitu: [2]

- Jumlah test case dapat dikurangi untuk mencapai pengujian yang wajar
- Test case dapat menunjukkan ada atau tidaknya kesalahan.
- Penguji tidak harus tahu coding.
- Programmer dan penguji keduanya independen satu sama lain.
- Lebih efektif daripada white box testing pada source code yang panjang.

Selain kelebihan-kelebihan tersebut, black box testing juga memiliki beberapa kekurangan, diantaranya: [2]

- Test case sulit dirancang tanpa spesifikasi yang jelas.
- Hanya sejumlah kecil input yang mungkin yang benar-benar dapat diuji.
- Beberapa bagian ujung belakang tidak diuji sama sekali.
- Peluang memiliki jalur yang tidak dikenal selama pengujian ini
- Peluang memiliki pengulangan tes yang sudah dilakukan oleh programmer

2.4 BVA(Boundary Value Analysis)

Pada Boundary Value Analysis, diuji nilai input apakah berada pada batasan atau di atas atau di bawah batasan. Pengalaman menunjukkan bahwa teknik ini memiliki peluang lebih tinggi untuk mendeteksi kesalahan [3].

Beberapa prinsip yang mendasari boundary value analysis (BVA) yaitu:[4]

1. Banyak kesalahan terjadi pada kesalahan masukan.
2. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input.
3. BVA merupakan komplemen dari equivalence partitioning. Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.

Pengujian dengan menggunakan metode BVA dapat dikerjakan secara generic. Bentuk dasar implementasi BVA adalah untuk menjaga agar satu variable berada pada nilai nominal (normal atau rata-rata) dan mengijinkan variable lain diisikan dengan nilai ekstrimnya [4]. Nilai yang digunakan untuk menguji keekstriman dapat dilihat pada Tabel I:

TABLE I. KATEGORI KEEKSTRIMAN DATA

Kategori	Keterangan
Min-	Tepat di bawah Minimum
Min	Minimum
Min+	Tepat di atas Minimum
Nom	Rata-rata
Max-	Tepat di bawah Maksimum
Max	Maksimum
Max+	Tepat di atas Maksimum

2.5 Decision Table Testing

Pada pengujian software dengan mengunakan metode decision table testing sejumlah kombinasi input dipergunakan untuk menentukan output yang seharusnya. Decision table testing digunakan untuk pengujian relasi logika yang kompleks [3].

Decision Table Testing menjadi penting ketika diperlukan untuk menguji kombinasi yang berbeda. Teknik ini juga membantu dalam cakupan pengujian yang lebih baik untuk logika bisnis yang kompleks.

Dalam Software Engineering, jika untuk setiap set nilai input menghasilkan perilaku sistem yang bisa berbeda, teknik Boundary Value Analysis tidak efektif digunakan. Dalam hal ini, pengujian dengan menggunakan teknik Decision Table Testing adalah pilihan yang lebih baik. Teknik ini dapat memastikan cakupan yang baik, dan representasi yang sederhana sehingga mudah untuk ditafsirkan dan digunakan [5].

Contoh Decision Table Testing untuk pengujian screen login seperti ditunjukkan pada Gbr. 3.

Gambar 3 Screen Login

Kondisinya sederhana, jika user menginput username berupa email dan password yang benar, maka sistem akan melakukan redirect ke homepage. Jika salah satu input salah maka akan ditampilkan pesan kesalahan [5].

Dari sini dapat dibuat Decision Table sebagai berikut:

TABLE II. DECISION TABLE SCREEN LOGIN

Kondisi	Aturan 1	Aturan 2	Aturan 3	Aturan 4
Username (T/F)	F	T	F	T
Password (T/F)	F	F	T	T
Output (E/H)	E	E	E	H

Keterangan:

T : Username/Password benar

F : Username/Password salah

E : Pesan error ditampilkan

H : Homepage ditampilkan

Interpretasi:

Aturan 1 : Username dan Password salah keduanya, tampil pesan kesalahan

Aturan 2 : Username benar, tetapi Password salah, tampil pesan kesalahan

Aturan 3 : Username salah, tetapi Password benar, tampil pesan kesalahan

Aturan 4 : Username dan Password benar keduanya, navigasi ke Homepage.

3. METODOLOGI PENELITIAN

3.1 Obyek Penelitian

Pada penelitian ini yang menjadi obyek penelitian adalah software yang digunakan untuk pengelolaan dana pensiun karyawan. Penelitian ini bertujuan untuk melihat penerapan metode pengujian software, sehingga tidak dilakukan pengujian untuk semua form yang ada. Dipilih salah satu form yang akan diuji dengan menggunakan metode Boundary Value Analysis dan metode Decision Table Testing.

Form yang dipilih untuk diuji adalah form Pengajuan Klaim Manfaat Pensiun Berkala. Form ini cukup penting karena dipergunakan untuk menghitung besar manfaat pensiun berkala. Hasil perhitungan ini akan dijadikan dasar untuk melakukan pembayaran manfaat pensiun.

3.2 Kerangka Berpikir

Berdasarkan studi literatur, untuk mendapatkan software yang berkualitas sangatlah penting untuk melakukan pengujian software. Ada banyak strategi yang dapat dipergunakan untuk melakukan pengujian software.

Salah satun strategi yang dapat dipergunakan adalah Black Box Testing. Pada strategi Black Box Testing juga terdapat beberapa metode. Pada penelitian ini dipergunakan metode Boundary Value Analysis dan Decision Table Testing untuk menguji salah satu form dari software yang digunakan untuk pengelolaan dana pensiun suatu perusahaan.

Untuk proses yang berdasarkan batasan nilai, diuji dengan menggunakan metode Boundary Value Analysis. Sedangkan untuk proses yang berdasarkan kombinasi input diuji dengan menggunakan metode Decision Table Testing.

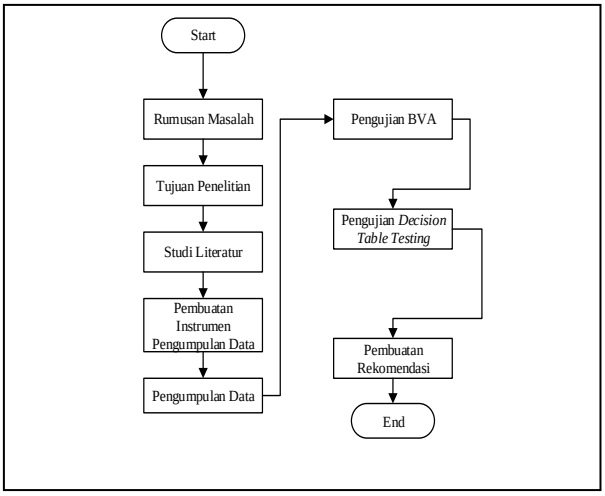
Dari hasil pengujian maka dapat diambil kesimpulan apakah form yang diuji sudah memenuhi kebutuhan atau belum.

3.3 Tahapan Penelitian

Tahapan penelitian pada Gbr. 4, dimulai dengan mendefinisikan masalah beserta batasan masalahnya. Lalu menentukan tujuan dan manfaat dari penelitian. Setelah itu melakukan studi literatur dan mengambil data yang akan diteliti.

Data yang terkumpul dari wawancara dipergunakan untuk membuat skenario pengujian. Data yang terkumpul dari hasil pengamatan pada saat pengujian form diolah dengan menggunakan analisa deskriptif untuk pengambilan kesimpulan dan rekomendasi.

.



Gbr. 3 Tahapan Penelitian

3.4 Metode Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan cara wawancara untuk memahami proses bisnis software yang akan diuji, dan pengamatan terhadap proses pengujian software. Dari data hasil wawancara disusun aturan pengujian software, sedangkan dari data pengamatan proses pengujian software dipergunakan untuk mengambil kesimpulan hasil pengujian dan membuat rekomendasi.

Pengolahan data dilakukan dengan analisa deskriptif. Data hasil wawancara diolah untuk membuat aturan-aturan yang akan dipergunakan untuk pengujian dengan metode Boundary Value

Analysis maupun Decision Table Testing. Sedangkan data hasil pengamatan pengujian software diperguankan untuk menganalisa hasil pengujian dan membuat rekomendasi atas proses pengujian software.

4. HASIL PENELITIAN

4.1 Pengumpulan Data

Wawancara dilakukan kepada Business Analyst untuk mengetahui proses bisnis Pengajuan Klaim Pensiun Manfaat Berkala. Berdasarkan hasil wawancara tersebut dapat dibuat aturan pengujiam form.

1) Proses Bisnis Pengajuan Klaim Manfaat Pensiun Berkala

Ketika ada peserta yang mengajukan klaim manfaat pensiun berkala, akumulasi dana pensiun sudah dihitung secara otomatis oleh sistem lain. Dari hasil perhitungan tersebut peserta dapat memilih Term Annuity dan Porsi Cadangan, yang akan menentukan manfaat pensiun berkala yang diterima setiap bulannya.

Selain itu juga dipilih jenis PTKP (Pendapatan Tidak Kena Pajak) peserta, apakah termasuk berkeluarga tanpa tanggungan, berkeluarga dengan satu tanggungan, berkeluarga dengan dua tanggungan, dan sebagainya. Pilihan ini berdasarkan ketentuan dari Departemen Keuangan yang akan digunakan untuk menghitung pajak penghasilan dari manfaat pensiun. Lalu diinput data rekening bank peserta yang akan digunakan untuk menerima manfaat pensiun setiap bulannya. Informasi rekening bank terdiri dari Bank, No Rekening, Pemilik Rekening, Cabang, dan Kota Bank. Petugas juga dapat mengisi Tanggal Jatuh Tempo Bayar, yaitu tanggal pembayaran manfaat pensiun setiap bulannya.

Tampilan form pengajuan Klaim Pensiun Manfaat Berkala dapat dilihat pada Gbr. 4

Gbr. 4 Form Pengajuan Klaim Manfaat Pensiun Berkala

2) Aturan Penginputan dan Submit Form

Berdasarkan tampilan form pada Gbr. 4 terdapat beberapa input. Input ada yang divalidasi berdasarkan rentang nilai yang diijinkan dan ada yang divalidasi berdasarkan mandatori atau tidak.

Input yang divalidasi berdasarkan rentang nilai yang diijinkan akan diuji dengan menggunakan metode *Boundary Value Analysis*. Input yang terkait dengan pengujian dengan menggunakan metode *Boundary Value Analysis* beserta batasan nilainya dapat dilihat pada Tabel III

TABLE III. BATASAN NILAI INPUT

No	Input	Batasan Min	Batasan Max
R1	Tgl Jatuh Tempo Bayar	Tanggal Sistem	-
R2	No Rekening	8 huruf	15 huruf
R3	Pemilik Rekening	3 huruf	100 huruf
R4	Cabang Bank	3 huruf	30 huruf
R5	Kota Bank	3 huruf	30 huruf

Input yang divalidasi berdasarkan mandatori atau tidak akan diuji dengan menggunakan metode *Decision Table Testing*. Input tersebut akan divalidasi pada saat form disubmit. Aturan validasi yang akan digunakan untuk melakukan pengujian dengan metode *Decision Table Testing* dapat dilihat pada Tabel IV.

TABLE IV. ATURAN VALIDASI SUBMIT FORM

Kondisi Input			Output
PTKP	Term Annuity	Porsi Cadangan	
F	F	F	E
F	T	T	E
F	F	T	E
F	T	F	E
T	F	F	E
T	F	T	E
T	T	F	E
T	T	T	S

Keterangan:

T : Input diisi

F : Input tidak diisi

E : Pesan error ditampilkan

S : Perhitungan manfaat pensiun ditampilkan

Interpretasi:

Aturan 1 : PTKP, Term Annuity, dan Porsi Cadangan ketiganya tidak diisi, tampil pesan kesalahan

Aturan 2 : PTKP tidak diisi, Term Annuity dan Porsi Cadangan diisi, tampil pesan kesalahan

Aturan 3 : PTKP dan Term Annuity tidak diisi, Porsi Cadangan diisi, tampil pesan kesalahan

Aturan 4 : PTKP dan Porsi Cadangan tidak diisi, Term Annuity diisi, tampil pesan kesalahan

Aturan 5 : PTKP diisi, Term Annuity dan Porsi Cadangan tidak diisi, tampil pesan kesalahan

Aturan 6 : PTKP dan Porsi Cadangan, Term Annuity tidak diisi, tampil pesan kesalahan

Aturan 7 : PTKP dan Term Annuity diisi, Porsi Cadangan tidak diisi, tampil pesan kesalahan

Aturan 8 : PTKP, Term Annuity, dan Porsi Cadangan diisi ketiganya, tampilkan hasil perhitungan manfaat pensiun.

4.2 Hasil Pengujian dengan Metode *BVA (Boundary Value Analysis)*

Untuk setiap input yang diuji disiapkan data berdasarkan keekstriman data, yaitu kategori Min, Min-, Min+, Nom, Max, Max-, Max+

1) Hasil Pengujian Input Tanggal Jatuh Tempo

Pada input Tanggal Jatuh Tempo tidak terdapat batasan maksimum, sehingga hanya terdapat batasan minimum yaitu tanggal sistem. Pada penelitian ini tanggal sistem diset 1 September 2019. Hasil pengujian terlihat pada Tabel V.

TABLE V. HASIL UJI ATURAN R1

Data Contoh	Harapan Hasil	Hasil	Kesimpulan
30/08/2019	Error	Error	Lolos
01/09/2019	Error	Error	Lolos
02/09/2019	Sukses	Sukses	Lolos

Untuk aturan R1, batasan Tanggal Jatuh Tempo, validasi input lolos uji.

2) Hasil Pengujian Input No Rekening

Pada input No Rekening terdapat batasan minimum 8 huruf dan batasan maksimum 15 huruf. Hasil pengujian terlihat pada Tabel VI.

TABLE VI.

TABLE VII. HASIL UJI ATURAN R2

Data Contoh	Harapan Hasil	Hasil	Kesimpulan
7 huruf	Error	Sukses	Gagal
8 huruf	Sukses	Sukses	Lolos
9 huruf	Sukses	Sukses	Lolos
11 huruf	Sukses	Sukses	Lolos
14 huruf	Sukses	Sukses	Lolos
15 huruf	Sukses	Sukses	Lolos
16 huruf	Error	Sukses	Gagal

Untuk aturan R2, batasan No Rekening, validasi input masih gagal.

3) Hasil Pengujian Input Pemilik Rekening

Pada input Pemilik Rekening terdapat batasan minimum 3 huruf dan batasan maksimum 100 huruf. Hasil pengujian terlihat pada Tabel VII.

TABLE VIII. HASIL UJI ATURAN R3

Data Contoh	Harapan Hasil	Hasil	Kesimpulan
2 huruf	Error	Sukses	Gagal
3 huruf	Sukses	Sukses	Lolos
4 huruf	Sukses	Sukses	Lolos
50 huruf	Sukses	Sukses	Lolos
99 huruf	Sukses	Sukses	Lolos
100 huruf	Sukses	Sukses	Lolos
101 huruf	Sukses	Sukses	Lolos

Untuk aturan R3, batasan Pemilik Rekening, validasi input masih gagal.

4) Hasil Pengujian Input Cabang Bank

Pada input Cabang Bank terdapat batasan minimum 3 huruf dan batasan maksimum 30 huruf. Hasil pengujian terlihat pada Tabel VIII.

TABLE IX. HASIL UJI ATURAN R4

Data Contoh	Harapan Hasil	Hasil	Kesimpulan
2 huruf	Error	Error	Lolos
3 huruf	Sukses	Sukses	Lolos

Data Contoh	Harapan Hasil	Hasil	Kesimpulan
4 huruf	Sukses	Sukses	Lolos
15 huruf	Sukses	Sukses	Lolos
29 huruf	Sukses	Sukses	Lolos
30 huruf	Sukses	Sukses	Lolos
31 huruf	Sukses	Sukses	Lolos

Untuk aturan R4, batasan Cabang Bank, validasi input lolos uji.

5) Hasil Pengujian Input Kota Bank

Pada input Kota Bank terdapat batasan minimum 3 huruf dan batasan maksimum 30 huruf. Hasil pengujian terlihat pada Tabel IX.

TABLE X. HASIL UJI ATURAN R5

Data Contoh	Harapan Hasil	Hasil	Kesimpulan
2 huruf	Error	Error	Lolos
3 huruf	Sukses	Sukses	Lolos
4 huruf	Sukses	Sukses	Lolos
15 huruf	Sukses	Sukses	Lolos
29 huruf	Sukses	Sukses	Lolos
30 huruf	Sukses	Sukses	Lolos
31 huruf	Sukses	Sukses	Lolos

Untuk aturan R5, batasan Kota Bank, validasi input lolos uji.

Dapat dilihat bahwa pada pengujian form dengan menggunakan metode *Boundary Value Analysis (BVA)* ditemukan bahwa sudah lolos uji untuk input Tanggal Jatuh Tempo, Kota Bank, dan Cabang Bank. Namun masih terdapat *bug* pada validasi input No Rekening dan Pemilik Rekening.

4.3 Hasil Pengujian dengan *Metode Decision Table Testing*

Pada pengujian ini dilakukan kombinasi input untuk menentukan output yang diharapkan. Hasil pengujian dapat dilihat pada Tabel X

TABLE XI. HASIL PENGUJIAN DECISION TABLE TESTING

Kondisi Input			Harapan Output	Output	Kesimpulan
PTKP	Term Annuity	Porsi Cadangan			
F	F	F	E	E	Lolos
F	T	T	E	E	Lolos
F	F	T	E	E	Lolos
F	T	F	E	E	Lolos
T	F	F	E	E	Lolos
T	F	T	E	E	Lolos
T	T	F	E	E	Lolos
T	T	T	S	S	Lolos

Dapat dilihat bahwa pada pengujian form dengan menggunakan metode *Decision Table Testing*, form yang diuji sudah lolos uji.

4.4 Rekomendasi

Pengujian software dengan menggunakan metode Boundary Value Analysis dan Decision Table Testing dapat digunakan untuk menguji apakah software yang dibuat sudah sesuai dengan kebutuhan proses bisnis. Untuk yang masih belum lolos uji perlu dilakukan perbaikan sehingga sesuai dengan ketentuan proses bisnis.

Dapat dilakukan penelitian lanjutan untuk menggunakan metode lainnya atau menguji form lain yang memiliki proses bisnis yang berbeda. Dengan demikian software yang dibuat betul-betul telah memenuhi kebutuhan proses bisnis.

5. KESIMPULAN

Pengujian software dengan menggunakan metode *Boundary Value Analysis* dapat menguji apakah nilai input yang dimasukkan ke sistem berada dalam rentang batas yang diijinkan oleh proses bisnis. Pada contoh form yang diuji pada penelitian ini masih terdapat *bug* yang perlu diperbaiki.

Dengan menggunakan metode *Decision Table Testing*, pengujian software dapat dilakukan untuk memvalidasi perilaku sistem dalam menampilkan output berdasarkan kombinasi input yang diberikan. Contoh form yang diuji pada penelitian ini sudah memenuhi ketentuan proses bisnis yang berlaku.

Dapat dilakukan penelitian lanjutan untuk menguji software menggunakan metode yang berbeda, atau menguji form lain yang memiliki proses bisnis yang berbeda.

DAFTAR PUSTAKA

[1] A. Anitha, "A Brief Overview of Software Testing Techniques and Metrics," International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 12, pp. 4655-4659, Desember 2013.

[2] Abhijit. A. Sawant, Pranit. H. Bari dan P. M. Chawan, "Software Testing Techniques and Strategies," International Journal of Engineering Research and Applications (IJERA), vol. 2, no. 3, pp. 980-986, May-Jun 2012.

[3] Meenu dan Navita, "Study and Analysis of Software Testing," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 12, pp. 6674-6678, Desember 2015.

[4] M. S. Mustaqbal, R. F. Firdaus dan H. Rahmadi, " Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)," Jurnal Ilmiah Teknologi Informasi Terapan (JITTER) Universitas Widyatama, vol. 1, no. 3, pp. 31-36, Agustus 2015.

[5] “Decision Table Testing: Learn with Example,” [Online]. Available: https://www.guru99.com/decision-table-testing.html. [Diakses 05 Maret 2019].