

PENGUJIAN BLACK BOX PADA APLIKASI DATABASE PERGURUAN TINGGI DENGAN TEKNIK EQUIVALENCE PARTITIONS

Hamzah Raihan Ikhsanul Fikri¹⁾, Apriade Voutama²⁾

^{1,2)}Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

e-mail: 2010631250052@student.unsika.ac.id¹⁾, apriade.voutama@staff.unsika.ac.id²⁾

Abstrak : Di era digital ini, perkembangan teknologi informasi semakin menawarkan berbagai jenis inovasi yang ditujukan untuk kenyamanan setiap pengguna. Aplikasi database universitas adalah aplikasi yang digunakan untuk mengelola data mahasiswa, fakultas, mata kuliah, jurusan dan nilai mahasiswa. Ada 2 metode pengujian aplikasi yaitu pengujian white box dan black box. Untuk pengujian perangkat lunak ini akan dilakukan pengujian menggunakan metode black box dengan teknik titik ekuivalen. Pengujian Black Box tidak memerlukan kode sumber. Partisi ekuivalen adalah suatu metode pengujian berdasarkan performansi data input dalam suatu sistem aplikasi data performansi, dimana setiap menu input diuji dan juga dikelompokkan sesuai fungsinya, valid atau tidak. Pengujian dilakukan dengan alasan untuk mengetahui apa saja permasalahan yang terdapat pada aplikasi sebelum digunakan oleh staff perguruan tinggi. Tujuan dari metode ini adalah untuk mendeteksi kesalahan aplikasi sehingga dapat diperbaiki lebih cepat. Hasil pengujian terdapat 35 kasus uji dari aplikasi database universitas dan 4 kasus uji yang diperoleh tidak sesuai dengan yang diharapkan. Meskipun ada 31 kasus uji yang berhasil, ada hal-hal yang perlu diperbaiki dari output yang dihasilkan oleh aplikasi. Ada beberapa hal yang perlu diperbaiki pada aplikasi, seperti tidak ada pesan berupa alert saat melakukan penambahan, mengubah, dan menghapus data, dan sistem dapat menerima request meskipun data tidak terisi.

Kata Kunci—Black Box Testing, Database, Equivalence Partition, Perangkat Lunak, White Box Testing

Abstract : In this digital era, the development of information technology increasingly offers various types of innovations aimed at the convenience of every user. The university database application is an application that is used to manage student data, faculties, courses, majors and student grades. There are 2 application testing methods, namely white box testing and black box testing. For testing this software will be tested using the black box method with the equivalence point technique. Black Box testing does not require source code. Equivalent partitioning is a test method based on the performance of input data in a performance data application system, where each input menu is tested and also grouped according to its function, valid or not. The purpose of this method is to detect application errors so that they can be fixed more quickly. The test results contained 35 test cases from the university database application and 4 test cases obtained were not as expected. Although there are 31 successful test cases, there are things that need to be improved from the output generated by the application. There are several things that need to be improved in the application, such as no alert message when adding, changing, and deleting data, and the system can accept requests even though the data is not filled in.

Keywords—Black Box Testing, Database, Equivalence Partition, Software, White Box Testing

I. PENDAHULUAN

PENGUJIAN dalam suatu program harus dilakukan, hal ini penting karena untuk mengecek semua kesalahan yang ada pada program agar tidak terjadi kerugian akibat kesalahan tersebut, sehingga sangat perlu dilakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan [1]. Di era digital ini, perkembangan teknologi informasi semakin menawarkan berbagai jenis inovasi yang ditujukan untuk kenyamanan setiap pengguna [2]. Hasilnya, banyak sekali aplikasi yang ada untuk memudahkan kegiatan masyarakat. Namun, biasanya aplikasi tersebut memiliki *bug* dan *error* yang harus diperbaiki sebelum perilisasi. Maka dari itu dalam penelitian ini akan menguji salah satu aplikasi yang memudahkan dalam kegiatan penyimpanan data perguruan tinggi.

Pada umumnya ada dua metode yang dilakukan untuk pengujian aplikasi. Metode pertama adalah metode white box. Metode white box digunakan untuk menguji bagaimana aplikasi bekerja, tes ini dilakukan untuk menunjukkan tingkat bagaimana aplikasi bekerja sesuai dengan prosedur dan spesifikasi. Metode kedua adalah black box. Metode black box digunakan untuk pengujian apakah aplikasi tersebut dapat memenuhi fungsinya dengan baik [3].

Black Box Testing memiliki keuntungan karena tidak memerlukan *source code*, sehingga tidak memerlukan instrumentasi dan ketersediaan *source code*. Di sisi lain, orang dapat berhipotesis bahwa mengakses *source code* menggunakan white box testing dapat meningkatkan cakupan *source code* dan meningkatkan laporan bug awal. White Box Testing juga ditemukan mahal, dan yang menggunakan informasi cakupan dari rilis sebelumnya dapat secara efektif mengurangi prioritas di beberapa rilis. Teknik memprioritaskan kasus uji white box bisa jadi: tidak berlaku ketika tidak ada *source code* atau instrumentasi tidak memungkinkan, jadi penguji mungkin tidak punya pilihan selain menggunakan Black Box Testing. Oleh karena itu, pada studi kasus kali ini akan dilakukan untuk menguji aplikasi database perguruan tinggi menggunakan black box testing.

Aplikasi database perguruan tinggi pada Universitas Singaperbangsa Karawang merupakan aplikasi yang digunakan dalam pengelolaan data mahasiswa, dosen, mata kuliah, jurusan, dan nilai mahasiswa. Aplikasi ini dikembangkan menggunakan bahasa pemrograman java. Database pada aplikasi ini memakai MySQL sebagai server penyimpanan data. Aplikasi ini diharapkan dapat membuat proses input data semakin efisien. Dengan adanya aplikasi database ini setidaknya dapat mengurangi penyimpanan dokumentasi secara tradisional yang biasanya menggunakan kertas. Aplikasi database ini masih dalam tahap pengembangan. Aplikasi ini memiliki fitur yang berguna untuk produktivitas dosen. Dengan membuat kegiatan dosen dalam menulis dokumentasi menjadi lebih mudah.

Fitur dan fungsi yang terdapat pada aplikasi ini diantaranya adalah 1) input data mahasiswa, 2) input data dosen, 3) input data jurusan, 4) input data nilai, 5) input data mata kuliah, 6) hapus data, 7) dan tampilan menu data. Kelebihan pada aplikasi ini adalah aplikasi ini dapat membuat laporan berupa file pdf. Maka pengguna tidak perlu membuat secara manual laporan dari data yang sudah diinputkan. Dengan adanya fitur-fitur pada aplikasi tersebut, maka perlu dilakukan *testing* untuk menguji dan memastikan fungsi yang disediakan bekerja sesuai harapan agar tidak terjadi kesalahan pada input maupun output yang dapat membuat informasi pada data keliru atau tidak valid. Pada pengujian ini akan dilakukan pengujian pada semua fitur yang tersedia di aplikasi database perguruan tinggi. Pengujian ini dilakukan dengan menggunakan teknik equivalence partition.

Partisi ekuivalen adalah suatu metode pengujian berdasarkan performansi data input dalam suatu sistem aplikasi informasi data performansi, dimana setiap menu input diuji dan juga dikelompokkan berdasarkan fungsinya, valid atau tidak [4]. Pengujian Equivalence Partitioning membagi domain input program ke dalam kelas-kelas data, di mana kasus uji yang dapat dilatih berdasarkan input dan output suatu komponen dipartisi ke dalam kelas-kelas. Hal ini dilakukan untuk mendapatkan dataset berupa dokumentasi pengujian dengan metode Equivalence Partitions dan nilai tingkat efektifitas metode Equivalence Partitions [5]. *Test case* akan diperiksa dengan metode equivalence partitioning, langkah ini dilakukan untuk mendapatkan kumpulan data, kemudian kumpulan data yang berisi data *test case* akan dihitung benar atau tidaknya [6]. Pada hasil pengujian terdapat tabel rancangan *Test Case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak rancangan *test case* berdasarkan Equivalence Partitions [7].

Hasil pengujian pada aplikasi database perguruan tinggi Universitas Singaperbangsa Karawang ini diharapkan dapat membuat aplikasi tersebut menjadi optimal dengan fitur yang berfungsi semestinya tanpa adanya kesalahan pada sistem sehingga dapat digunakan oleh staff perguruan tinggi Universitas Singaperbangsa Karawang.

II. TINJAUAN PUSTAKA

A. Database

Pada dasarnya database dapat dipahami sebagai tempat penyimpanan data yang menggantikan sistem konvensional berupa file dokumen [8]. Database atau basis data didefinisikan sebagai kumpulan data yang saling terkait dan deskriptif yang dibuat sebagai sarana untuk memenuhi tugas dalam penyimpanan suatu informasi data suatu organisasi [9]. Tidak seperti sistem file yang menyimpan data secara terpisah, data dalam database disimpan secara terintegrasi. [8]. Database memudahkan dalam mengelola file, sehingga mudah untuk mencari, menyimpan, dan menghapus sebuah informasi. Ada beberapa jenis server database seperti MySQL, MariaDB dan PostgreSQL. .

B. Black Box Testing

Pengujian black box adalah metode pengujian perangkat lunak yang terutama memeriksa spesifikasi perangkat lunak yang sedang dikembangkan. Pengujian black box dapat menemukan beberapa hal seperti fungsionalitas yang salah atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antarmuka, kesalahan kinerja dan inisialisasi, dan akhir [10]. Black Box Testing, juga dikenal sebagai pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Dengan kata lain, pengujian kotak hitam memberi insinyur perangkat lunak satu set kondisi input yang secara sempurna memenuhi semua persyaratan fungsional program. Black box testing mencoba menemukan kesalahan dalam kategori berikut:

1. Kesalahan atau fungsi yang hilang.
2. Error pada interface.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan operasional atau kinerja.
5. Kesalahan inisialisasi dan terminasi. [11]

Metodologi pengujian perangkat lunak black box yang paling umum adalah: [12]

1. Boundary Value Analysis
Sistem mungkin gagal pada batasnya. Karena ada kesalahan yang dapat dilakukan programmer pada batas kelas ekivalensi. Oleh karena itu, teknik ini berfokus pada edge atau nilai yang dipilih dengan batasan ekstrim.
2. Equivalence Partitioning
Teknik ini membantu mengurangi jumlah kasus uji. Ini pada dasarnya bekerja pada dan di atasnya dengan membagi domain input program menjadi kelas kesetaraan berdasarkan nilai input. Kasus uji yang dihasilkan dari kelas ekivalensi ini diturunkan dari domain input.
3. Orthogonal Array Testing
Orthogonal array testing adalah metode uji statistik. Ini digunakan ketika domain input sangat kecil dan membantu mengurangi jumlah kombinasi pengujian. Variabel diwakili oleh kolom dan kasus uji oleh baris.
4. Fuzzing
Fuzzing adalah teknik pengujian kotak hitam yang dikembangkan pada tahun 1989 oleh Barton Miller di University of Wisconsin. Teknik ini didasarkan pada pemberian masukan acak ke aplikasi. Dengan menggunakan teknik pengujian fuzz untuk memasukkan error/data semi-cacat, kesalahan implementasi dapat ditemukan pada pengujian fuzz sesi otomatis atau semi-otomatis.
5. Graph Based Testing
Graph based testing adalah metode pengujian kotak hitam yang dimulai dengan pembuatan grafik. Grafik dibuat dari modul input. Modul input diberi pengidentifikasi. Grafik menunjukkan hubungan antara akibat dan sebab.
6. All Pair Testing

All pair testing adalah teknik untuk menguji semua kemungkinan pasangan diskrit untuk parameter masukan setiap pasangan untuk mencakup semua pasangan yang memerlukan serangkaian kasus uji untuk mengeksekusi kombinasi.

7. State Transitions Diagrams atau State Graphs

State Graph merupakan teknik untuk menangkap berbagai jenis persyaratan sistem dan mendokumentasikan desain sistem internal. Alat ini juga digunakan untuk pengujian mesin negara dan navigasi GUI (antarmuka pengguna grafis).

C. Equivalence Partition

Metode Partisi Ekuivalen merupakan metode yang digunakan untuk pengujian Black Box untuk mempartisi atau membagi domain input program menjadi kelas-kelas data sehingga dapat diperoleh *test case* [13]. Pengujian fungsional untuk aplikasi database perguruan tinggi dilakukan dengan cara menguji fungsional menggunakan metode black box dengan teknik partisi ekuivalen. Tujuannya untuk menemukan kesalahan pada sistem aplikasi database perguruan tinggi yang dikembangkan. Pengujian dilakukan dengan membagi domain input ke dalam kelas-kelas sehingga diperoleh *test case* pada aplikasi [14].

D. Java

Java adalah bahasa pemrograman yang digunakan untuk pengembangan aplikasi berupa mobile, komputer, dan website. Bahasa pemrograman Java mempunyai bagian yang disebut class. Salah satu kelebihan Java adalah dapat dijalankan diberbagai sistem operasi, misalnya: Linux, Windows, Mac OS, Solaris, dan sebagainya [15].

E. MySQL

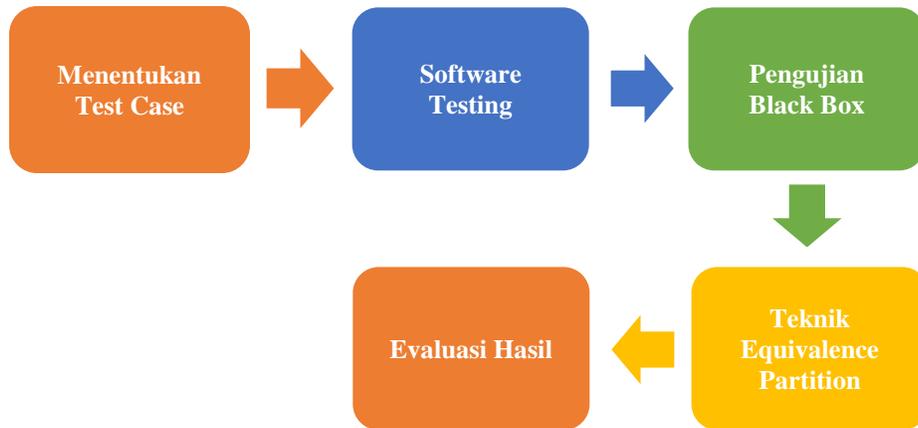
MySQL merupakan sistem manajemen database yang digunakan untuk mengelola data dalam website dan aplikasi. MySQL termasuk dalam tipe database yaitu RDBMS (Relational Database Management System). Pada MySQL, berisi satu atau lebih tabel. Sebuah tabel terdiri dari sejumlah baris dan setiap baris berisi satu atau lebih kolom [16]. MySQL merupakan program *multi-user* dan menggunakan perintah dasar SQL (*Structured Query Language*). MySQL yang sering digunakan adalah MySQL FreeSoftware yang menggunakan lisensi GNU/GPL (*General Public License*).

MySQL memiliki kelebihan dibanding database lain, diantaranya:

1. MySQL merupakan sebuah database yang *free*.
2. MySQL merupakan database dengan luas penyimpanan hingga *gigabyte* lebih.
3. MySQL merupakan *multi-user*.
4. MySQL mempunyai kecepatan yang sangat cepat dalam pembuatan tabel maupun mengubah tabel.

III. METODE PENELITIAN

Pada tahap ini akan dijelaskan metode penelitian yang digunakan untuk memperjelas langkah-langkah selanjutnya dalam penelitian ini, terutama dalam melakukan pengujian menggunakan teknik ekivalensi partisi [17]. Penelitian akan dilakukan dengan melakukan beberapa tahapan yaitu menentukan *test case*, menguji perangkat lunak, pengujian black box, teknik equivalence partition, dan evaluasi hasil. Tahapan ini dilakukan untuk memberikan alur yang jelas dalam melakukan penelitian. Alur penelitian bisa dilihat di gambar 1



Gambar 1. Tahapan pengujian dengan Teknik Equivalence Partition

A. Menentukan Test Case

Tahap awal dimulai dengan mempersiapkan perangkat lunak yang akan diuji dan meliputi perencanaan untuk menentukan jenis pengujian yang akan digunakan. Beberapa fitur atau fungsi perangkat lunak diuji terhadap *test case* yang disiapkan. Hal ini dilakukan untuk mendapatkan kumpulan data sebagai dokumen uji menggunakan metode partisi ekivalensi dan nilai efisiensi.

B. Software Testing

Pada tahap kedua ini, dilakukan pengujian perangkat lunak pada aplikasi database perguruan tinggi. Pengujian pada aplikasi tersebut akan menggunakan laptop sebagai alat untuk penggunaan aplikasi. Beberapa fitur atau fungsi yang akan diuji meliputi input data mahasiswa, input data dosen, input data jurusan, input data nilai, input data mata kuliah, hapus data, dan tampilan menu data.

C. Pengujian Black Box

Metode pengujian yang digunakan adalah pengujian black box yang berfokus pada pengujian rinci aplikasi database perguruan tinggi seperti fitur atau fungsionalitas dari setiap halaman aplikasi. Pengujian kotak hitam tidak memeriksa sisi kode sumber program, tetapi memeriksa fitur atau fungsionalitas program operasi. Oleh karena itu, fokus utama dari pengujian ini hanya pada informasi kapabilitas dan fungsionalitas untuk setiap jenis program. Pengujian kotak hitam cenderung dapat menemukan beberapa hal seperti fungsionalitas yang salah atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antarmuka, kinerja, dan inisialisasi.

D. Teknik Equivalence Partition

Pada tahapan ini, teknik equivalence partition digunakan ketika input dibagi menjadi beberapa kelas, yaitu dengan hasil yang valid dan tidak valid. Lalu *test case* dibuat berdasarkan hasil dari masing-masing kelas. Untuk melihat apakah input data valid atau tidak valid yaitu jika kondisi input membutuhkan nilai yang spesifik maka equivalence akan ditentukan.

E. Evaluasi Hasil

Pada tahap terakhir ini hal ini akan memberikan temuan ketika melakukan pengujian dan mengungkapkan kekurangan serta kelebihan berdasarkan pelaksanaan. Ditahap ini bisa menambahkan rekomendasi agar aplikasi bisa menjadi lebih baik lagi. Evaluasi hasil juga akan memberikan penjelasan secara detail apa dan mengapa aplikasi tersebut gagal atau berhasil melakukan fungsinya.

IV. HASIL DAN PEMBAHASAN

A. Menentukan Test Case

Awal penelitian dilakukan dengan menentukan test case terlebih dahulu. Dalam tahap ini akan menjelaskan apa saja yang akan diuji dan hasil yang diharapkan. Pengujian akan dilakukan pada aplikasi database perguruan tinggi. Pembuatan tabel pengujian akan berisikan *test case* yang akan dilakukan saat menjalankan black box testing untuk menguji apakah suatu program berjalan sesuai dengan spesifikasi persyaratan *Test case* dibuat untuk mengetahui apakah aplikasi database perguruan tinggi ini layak untuk digunakan oleh para dosen dalam metode penyimpanan data atau tidak. Tujuan dibuatnya *test case* ini yaitu untuk dilakukan agar diketahuinya apakah aplikasi database perguruan tinggi ini layak untuk digunakan para dosen dalam metode penyimpanan data [18]. *Test case* dapat dilihat pada tabel 1.

Tabel 1. Menentukan Test Case Pengujian Aplikasi Database

ID	Test Case	Hasil yang Diharapkan
ID-A01	Membuka menu data mahasiswa	Sistem menampilkan daftar data mahasiswa
ID-A02	Melakukan penambahan data tanpa mengisi field input data mahasiswa	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”
ID-A03	Menambahkan data mahasiswa dengan NIM “121212”, nama mahasiswa “Jacquin Ragus”, tanggal lahir “1 Januari 2000”, jurusan “Teknik Informatika”, alamat “Jakarta”	Sistem menerima request penambahan data yang sudah diinputkan dan menampilkan alert “Data berhasil diinputkan”
ID-A04	Melakukan penghapusan data mahasiswa “Hamzah Raihan”	Sistem menerima request penghapusan data dan menghapus data mahasiswa “Hamzah Raihan” dari database dan menampilkan alert “Data berhasil dihapus”
ID-A05	Melakukan perubahan data atau update data mahasiswa dengan mengubah NIM menjadi kosong	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field NIM kosong”
ID-A06	Melakukan perubahan data atau update data mahasiswa dengan mengubah NIM menjadi “1234567”	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”
ID-A07	Melakukan cetak laporan data mahasiswa	Sistem menerima request dan mencetak laporan NIM, nama mahasiswa, tanggal lahir, jurusan, dan alamat berupa file pdf
ID-B01	Melakukan klik pada menu data dosen	Sistem menerima request dan menampilkan data dosen
ID-B02	Melakukan penambahan data tanpa mengisi field input data dosen	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”
ID-B03	Menambahkan data dosen dengan NID “12”, nama dosen “Dadang Yusuf”, jenis kelamin “Laki-laki”, alamat “Jakarta”	Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”
ID-B04	Melakukan penghapusan	Sistem menerima request penghapusan data

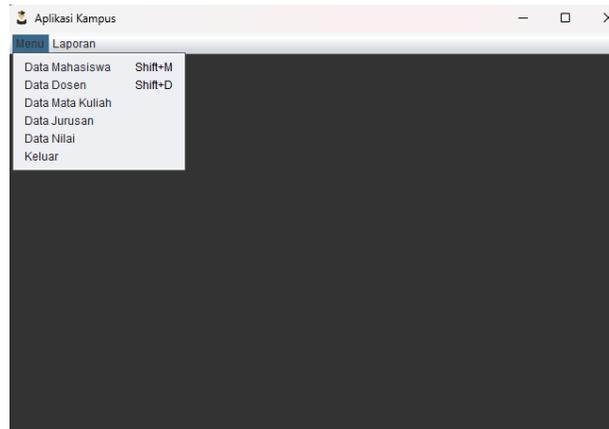
ID	Test Case	Hasil yang Diharapkan
	data dosen “Siska”	dan menghapus data dosen “Siska” dari database dan menampilkan alert “Data berhasil dihapus”
ID-B05	Melakukan perubahan data atau update data dosen dengan mengubah nama menjadi kosong	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field NIM kosong”
ID-B06	Melakukan perubahan data atau update data dosen dengan mengubah nama menjadi “Aji Subianto”	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”
ID-B07	Melakukan cetak laporan data dosen	Sistem menerima request dan mencetak laporan NID, nama dosen, jenis kelamin, alamat berupa file pdf
ID-C01	Membuka menu data mata kuliah	Sistem menampilkan daftar data mata kuliah
ID-C02	Melakukan penambahan data tanpa mengisi field input data mata kuliah.	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”
ID-C03	Menambahkan data mata kuliah dengan Kode “SI012”, nama mata kuliah “Testing dan implementasi Sistem Informasi”, SKS “3”	Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”
ID-C04	Melakukan penghapusan data mata kuliah “Rekayasa Perangkat Lunak”	Sistem menerima request penghapusan data dan menghapus data mata kuliah “Rekayasa Perangkat Lunak” dari database dan menampilkan alert “Data berhasil dihapus”
ID-C05	Melakukan perubahan data atau update data mahasiswa dengan mengubah SKS menjadi kosong	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field SKS kosong”
ID-C06	Melakukan perubahan data atau update data mata kuliah dengan mengubah SKS menjadi “4”	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”
ID-C07	Melakukan cetak laporan data mata kuliah	Sistem menerima request dan mencetak laporan Kode, mata kuliah, dan SKS berupa file pdf
ID-D01	Membuka menu data jurusan	Sistem menampilkan daftar data jurusan
ID-D02	Melakukan penambahan data tanpa mengisi field input data jurusan.	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”
ID-D03	Menambahkan data mata kuliah dengan Kode jurusan “10035”, nama jurusan “Teknik Informatika”, jumlah mahasiswa “1000”, gelar “S1”	Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”
ID-D04	Melakukan penghapusan data jurusan “Akuntansi”	Sistem menerima request penghapusan data dan menghapus data jurusan “Akuntansi” dari database dan menampilkan alert “Data

ID	Test Case	Hasil yang Diharapkan
		berhasil dihapus”
ID-D05	Melakukan perubahan data atau update data jurusan dengan mengubah jumlah mahasiswa menjadi kosong	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field jumlah mahasiswa kosong”
ID-D06	Melakukan perubahan data atau update data jurusan “Teknik Informatika” dengan mengubah jumlah mahasiswa menjadi “1100”	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”
ID-D07	Melakukan cetak laporan data jurusan	Sistem menerima request dan mencetak laporan Kode jurusan, nama jurusan, jumlah mahasiswa, dan gelar berupa file pdf
ID-E01	Membuka menu data nilai mahasiswa	Sistem menampilkan daftar data nilai mahasiswa
ID-E02	Melakukan penambahan data tanpa mengisi field input data jurusan.	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”
ID-E03	Menambahkan data nilai mahasiswa dengan nolap “1”, NIM “1234567”, nama mahasiswa “Hamzah Raihan”, mata kuliah “Testing dan Implementasi sistem informasi”, Absen “10”, Tugas “20”, UTS “80”, UAS “90”, dengan menghitung total, mutu, dan keterangan nilai secara otomatis	Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”
ID-E04	Melakukan penghapusan data nilai mahasiswa “Gilang Harsya”	Sistem menerima request penghapusan data dan menghapus data nilai mahasiswa “Gilang Harsya” dari database dan menampilkan alert “Data berhasil dihapus”
ID-E05	Melakukan perubahan data atau update data nilai mahasiswa dengan mengubah nilai UAS menjadi kosong	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field nilai UAS kosong”
ID-E06	Melakukan perubahan data atau update data nilai mahasiswa “Hamzah Raihan” dengan mengubah jumlah nilai UAS menjadi “20”	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”
ID-E07	Melakukan cetak laporan data nilai mahasiswa	Sistem menerima request dan mencetak laporan Nolap, NIM, nama mahasiswa, mata kuliah, absen, tugas, UTS, UAS, total, mutu, dan keterangan berupa file pdf.

B. Pengujian Test Case

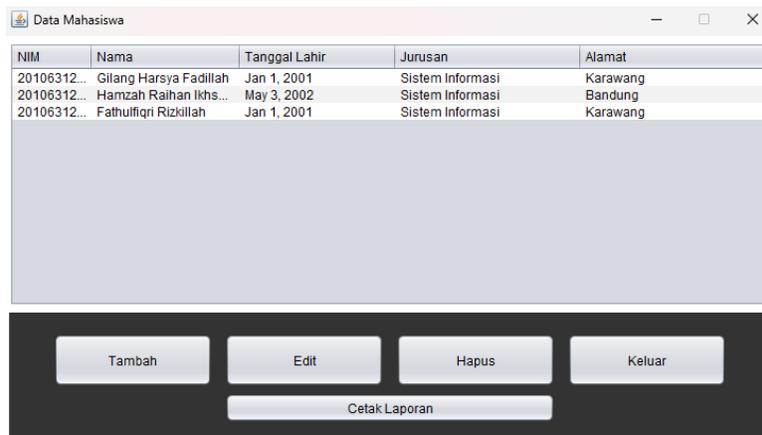
Pada tahapan ini dilakukan berdasarkan rancangan *test case* yang telah ditentukan pada tahap sebelumnya. Terdapat 35 *test case* yang akan digunakan sebagai acuan kasus apa saja yang akan diujikan secara

langsung pada aplikasi database perguruan tinggi. Pengujian input nilai yang dibagi menjadi lima *test case* pada fitur aplikasi yaitu fitur data mahasiswa, data dosen, data mata kuliah, data jurusan, dan data nilai mahasiswa. Pengujian ini akan diberi kode pada masing-masing *test case*, kode ID-A hingga ID-E. Hasil pengujian akan diberikan kode H-A01 dan seterusnya. Hasil pengujian akan menjelaskan apakah sistem tersebut berjalan sesuai yang diharapkan atau tidak. Pada tabel 1 ada beberapa *test case* untuk menguji terhadap sistem. Pengujian pertama, pada saat klik bagian menu data mahasiswa, sistem menampilkan menu daftar mahasiswa dengan button berupa tambah, edit, hapus, dan cetak laporan.



Gambar 2. Tampilan awal aplikasi

Gambar 2 menampilkan menu awal dari aplikasi. Menu dapat dilihat saat pertama kali membuka aplikasi.



Gambar 3. Tampilan menu data

Gambar 3 menampilkan menu data. Pada bagian ini, ada 4 *test case* yang akan dilakukan. Pertama, menggunakan fungsi tambah data. Kedua, menggunakan fungsi ubah atau update data. Ketiga, menggunakan fungsi hapus data. Keempat, menggunakan fungsi cetak laporan



Tambah Data

Tambah Data Mahasiswa

NIM

Nama

Tanggal Lahir ▼

Jurusan ▼

Alamat

Gambar 4. Tampilan form tambah data

Gambar 4 menampilkan menu untuk menambahkan data. Tampilan menu ini terdapat kolom input yang wajib untuk diisi.



Edit Data

Edit Data Mahasiswa

NIM

Nama

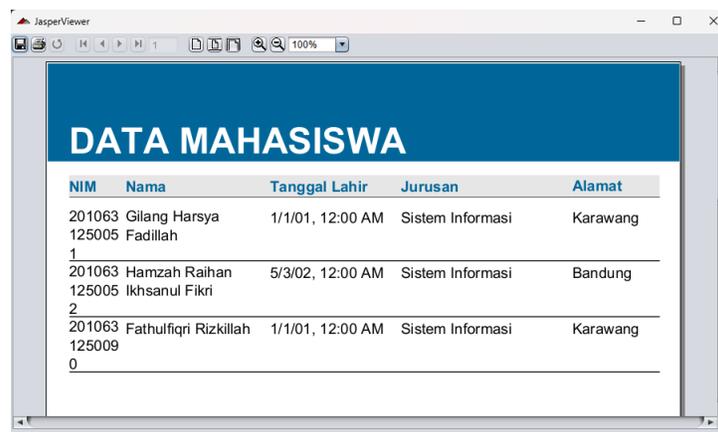
Tanggal Lahir ▼

Jurusan ▼

Alamat

Gambar 5. Tampilan ubah data atau update data

Gambar 5 menampilkan menu untuk mengubah data yang sebelumnya sudah di inputkan.



JasperViewer

DATA MAHASISWA

NIM	Nama	Tanggal Lahir	Jurusan	Alamat
201063	Gilang Harsya	1/1/01, 12:00 AM	Sistem Informasi	Karawang
125005	Fadillah	1		
201063	Hamzah Raihan	5/3/02, 12:00 AM	Sistem Informasi	Bandung
125005	Ikhsanul Fikri	2		
201063	Fathulfiqri Rizkillah	1/1/01, 12:00 AM	Sistem Informasi	Karawang
125009		0		

Page 1 of 1

Gambar 6. Tampilan cetak laporan

Gambar 6 menampilkan tampilan cetak laporan data. Pada tampilan cetak laporan, output yang dihasilkan berupa informasi mengenai data yang dapat di simpan menjadi PDF.

Tabel 2. Hasil Pengujian Software

ID	Test Steps	Hasil yang Diharapkan	Hasil Pengujian
H-A01	1. Buka aplikasi database perguruan tinggi. 2. Pilih menu data mahasiswa	Sistem menampilkan daftar data mahasiswa	Sistem menampilkan daftar data mahasiswa
H-A02	1. Tampilan input data mahasiswa 2. Input NIM : - 3. Input nama : - 4. Input tanggal lahir : - 5. Input jurusan : - 6. Input alamat : - 7. Klik button simpan	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”	Sistem menolak request penambahan data namun tidak menampilkan alert.
H -A03	1. Tampilan input data mahasiswa 2. Input NIM : 121212 3. Input nama : Jacquin Ragus 4. Input tanggal lahir : 1 Januari 2000 5. Input jurusan : Teknik Informatika 6. Input alamat : Jakarta 7. Klik button simpan	Sistem menerima request penambahan data yang sudah diinputkan dan menampilkan alert “Data berhasil diinputkan”	Sistem menerima request penambahan data yang sudah diinputkan namun tidak menampilkan alert.
H-A04	1. Tampilan menu data mahasiswa 2. Pilih data “Hamzah Raihan” 3. Klik hapus	Sistem menerima request penghapusan data dan menghapus data mahasiswa “Hamzah Raihan” dari database dan menampilkan alert “Data berhasil dihapus”	Sistem menerima request penghapusan data dan menghapus data mahasiswa “Hamzah Raihan” dari database namun tidak menampilkan alert
H-A05	1. Tampilan menu data mahasiswa 2. Pilih data “Hamzah Raihan” 3. Klik Edit 4. Input NIM : - 5. Klik simpan	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field NIM kosong”	Sistem menolak request perubahan data karena NIM tidak bisa diubah.
H-A06	1. Tampilan menu data mahasiswa 2. Pilih data “Hamzah Raihan” 3. Klik Edit 4. Input NIM : 1234567 5. Klik simpan	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”	Sistem menolak request perubahan data karena NIM tidak bisa diubah.
H-A07	1. Tampilan menu data mahasiswa 2. Klik cetak laporan	Sistem menerima request dan mencetak laporan NIM, nama mahasiswa, tanggal lahir, jurusan, dan alamat berupa file pdf	Sistem menerima request dan mencetak laporan NIM, nama mahasiswa, tanggal lahir, jurusan, dan alamat berupa file pdf
H-B01	1. Pilih menu data dosen	Sistem menampilkan daftar data dosen	Sistem menampilkan daftar data dosen
H-B02	1. Tampilan input data dosen 2. Input NID : - 3. Input nama : - 4. Input jenis kelamin : -	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”	Sistem menolak request penambahan data namun tidak menampilkan alert.

ID	Test Steps	Hasil yang Diharapkan	Hasil Pengujian
H-B03	<p>5. Input alamat : - 6. Klik button simpan</p> <p>1. Tampilan input data dosen 2. Input NID : 12 3. Input nama : Dadang Yusuf 4. Input jenis kelamin : Laki-laki 5. Input alamat : Jakarta 6. Klik button simpan</p>	<p>Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”</p>	<p>Sistem menerima request penambahan data namun tidak menampilkan alert</p>
H-B04	<p>1. Tampilan menu data dosen 2. Pilih data “Siska” 3. Klik hapus</p>	<p>Sistem menerima request penghapusan data dan menghapus data dosen “Siska” dari database dan menampilkan alert “Data berhasil dihapus”</p>	<p>Sistem menerima request penghapusan data dan menghapus data dosen “Siska” dari database namun tidak menampilkan alert</p>
H-B05	<p>1. Tampilan menu data dosen 2. Pilih data “Siska” 3. Klik edit 4. Input nama : - 5. Klik simpan</p>	<p>Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field NIM kosong”</p>	<p>Sistem menerima request perubahan data dan menampilkan tanpa input nama.</p>
H-B06	<p>1. Tampilan menu data dosen 2. Pilih data “Dadang Yusuf” 3. Klik edit 4. Input nama : Aji Subianto 5. Klik simpan</p>	<p>Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”</p>	<p>Sistem menerima request perubahan data dan namun tidak menampilkan alert</p>
H-B07	<p>1. Tampilan menu data dosen 2. Klik cetak laporan</p>	<p>Sistem menerima request dan mencetak laporan NID, nama dosen, jenis kelamin, alamat berupa file pdf</p>	<p>Sistem menerima request dan mencetak laporan NID, nama dosen, jenis kelamin, alamat berupa file pdf</p>
H-C01	<p>1. Pilih menu data mata kuliah</p>	<p>Sistem menampilkan daftar data mata kuliah</p>	<p>Sistem menampilkan daftar data mata kuliah</p>
H-C02	<p>1. Tampilan input data mata kuliah 2. Input kode : - 3. Input mata kuliah : - 4. Input SKS : - 5. Klik button simpan</p>	<p>Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”</p>	<p>Sistem menerima request penambahan data tanpa input field.</p>
H-C03	<p>1. Tampilan input data mata kuliah 2. Input kode : SI012 3. Input mata kuliah : Testing dan Implementasi Sistem Informasi. 4. Input SKS : 3 5. Klik button simpan</p>	<p>Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”</p>	<p>Sistem menerima request penambahan data namun tidak menampilkan alert.</p>
H-C04	<p>1. Tampilan menu data mata kuliah 2. Pilih data “Rekayasa Perangkat Lunak” 3. Klik hapus</p>	<p>Sistem menerima request penghapusan data dan menghapus data mata kuliah “Rekayasa Perangkat Lunak” dari database dan menampilkan alert “Data berhasil dihapus”</p>	<p>Sistem menerima request penghapusan data dan menghapus data mata kuliah “Rekayasa Perangkat Lunak” dari database dan namun tidak menampilkan alert</p>

ID	Test Steps	Hasil yang Diharapkan	Hasil Pengujian
H-C05	<ol style="list-style-type: none"> 1. Tampilan menu data dosen 2. Pilih data “Testing dan Impelementasi Sistem Informasi” 3. Klik edit 4. Input SKS : - 5. Klik simpan 	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field SKS kosong”	Sistem menerima request perubahan dengan field SKS kosong.
H-C06	<ol style="list-style-type: none"> 1. Tampilan menu data dosen 2. Pilih data “Testing dan Impelementasi Sistem Informasi” 3. Klik edit 4. Input SKS : 4 5. Klik simpan 	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”	Sistem menerima request perubahan data dan namun tidak menampilkan alert.
H-C07	<ol style="list-style-type: none"> 1. Tampilan menu data mata kuliah 2. Klik cetak laporan 	Sistem menerima request dan mencetak laporan Kode, mata kuliah, dan SKS berupa file pdf	Sistem menerima request dan mencetak laporan Kode, mata kuliah, dan SKS berupa file pdf
H-D01	<ol style="list-style-type: none"> 1. Pilih menu data mata jurusan 	Sistem menampilkan daftar data jurusan	
H-D02	<ol style="list-style-type: none"> 1. Tampilan input data jurusan 2. Input jurusan : - 3. Input jurusan : - 4. Input gelar : - 5. Input jumlah mahasiswa : - 5. Klik button simpan 	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”	Sistem menolak request penambahan data namun tidak menampilkan alert
H-D03	<ol style="list-style-type: none"> 1. Tampilan input data jurusan 2. Input jurusan : 10035 3. Input jurusan : Teknik Informatika 4. Input gelar : S1 5. Input jumlah mahasiswa : 1000 5. Klik button simpan 	Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”	Sistem menerima request penambahan data namun tidak menampilkan alert
H-D04	<ol style="list-style-type: none"> 1. Tampilan menu data jurusan 2. Pilih data “Akuntansi” 3. Klik hapus 	Sistem menerima request penghapusan data dan menghapus data jurusan “Akuntansi” dari database dan menampilkan alert “Data berhasil dihapus”	Sistem menerima request penghapusan data dan menghapus data jurusan “Akuntansi” dari database namun tidak menampilkan alert
H-D05	<ol style="list-style-type: none"> 1. Tampilan menu data jurusan 2. Pilih data “Teknik Informatika” 3. Klik edit 4. Input jumlah mahasiswa : - 5. Klik simpan 	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field jumlah mahasiswa kosong”	Sistem menolak request perubahan data namun tidak menampilkan alert
H-D06	<ol style="list-style-type: none"> 1. Tampilan menu data jurusan 2. Pilih data “Teknik 	Sistem menerima request perubahan data dan menampilkan alert “Ubah data	Sistem menerima request perubahan data namun tidak menampilkan alert

ID	Test Steps	Hasil yang Diharapkan	Hasil Pengujian
	Informatika” 3. Klik edit 4. Input jumlah mahasiswa : 1100 5. Klik simpan	berhasil dilakukan”	
H-D07	1. Tampilan menu data mata kuliah 2. Klik cetak laporan	Sistem menerima request dan mencetak laporan Kode jurusan, nama jurusan, jumlah mahasiswa, dan gelar berupa file pdf	Sistem menerima request dan mencetak laporan Kode jurusan, nama jurusan, jumlah mahasiswa, dan gelar berupa file pdf
H-E01	1. Pilih menu data nilai mahasiswa	Sistem menampilkan daftar data nilai mahasiswa	Sistem menampilkan daftar data nilai mahasiswa
H-E02	1. Tampilan form tambah data nilai mahasiswa 2. Input nolap : - 3. Input NIM : - 4. Input nama : - 5. Input mata kuliah : - 6. Input nilai absen : - 7. Input nilai UTS : - 8. Input nilai UAS : - 9. Input nilai tugas : - 10. total nilai 11. keterangan nilai 12. Klik button simpan	Sistem menolak request penambahan data dan menampilkan alert “Data gagal diinputkan, field harus diisi”	Sistem menolak request penambahan data namun tidak menampilkan alert
H-E03	1. Tampilan form tambah data nilai mahasiswa 2. Input nolap : 1 3. Input NIM : 1234567 4. Input nama : Hamzah Raihan 5. Input mata kuliah : Testing dan Impelementasi Sistem Informasi 6. Input nilai absen : 10 7. Input nilai UTS : 20 8. Input nilai UAS : 80 9. Input nilai tugas : 90 10. total nilai 11. keterangan nilai 12. Klik button simpan	Sistem menerima request penambahan data dan menampilkan alert “Data berhasil diinputkan”	Sistem menerima request penambahan data namun tidak menampilkan alert
H-E04	1. Tampilan menu data nilai mahasiswa 2. Pilih data “Gilang Harsya” 3. Klik button hapus	Sistem menerima request penghapusan data dan menghapus data nilai mahasiswa “Gilang Harsya” dari database dan menampilkan alert “Data berhasil dihapus”	Sistem menerima request penghapusan data dan menghapus data nilai mahasiswa “Gilang Harsya” dari database namun tidak menampilkan alert
H-E05	1. Tampilan form tambah data nilai mahasiswa 2. Input nolap : 1 3. Input NIM : 1234567 4. Input nama : Hamzah Raihan 5. Input mata kuliah : Testing dan Impelementasi Sistem Informasi	Sistem menolak request perubahan data dan menampilkan alert “Ubah data gagal dilakukan, field nilai UAS kosong”	Sistem menolak request perubahan data namun tidak menampilkan alert

ID	Test Steps	Hasil yang Diharapkan	Hasil Pengujian
H-E06	6. Input nilai absen : 10		
	7. Input nilai UTS : 20		
	8. Input nilai UAS : -		
	9. Input nilai tugas : 90		
	10. total nilai		
	11. keterangan nilai		
	12. Klik button simpan		
	1. Tampilan form tambah data nilai mahasiswa	Sistem menerima request perubahan data dan menampilkan alert “Ubah data berhasil dilakukan”	Sistem menerima request perubahan data namun tidak menampilkan alert
	2. Input nolap : 1		
	3. Input NIM : 1234567		
	4. Input nama : Hamzah Raihan		
	5. Input mata kuliah : Testing dan Impelementasi Sistem Informasi		
H-E07	6. Input nilai absen : 10		
	7. Input nilai UTS : 20		
	8. Input nilai UAS : 20		
	9. Input nilai tugas : 90		
	10. total nilai		
	11. keterangan nilai		
	12. Klik button simpan		
	1. Tampilan menu data mata kuliah	Sistem menerima request dan mencetak laporan Nolap, NIM, nama mahasiswa, mata kuliah, absen, tugas, UTS, UAS, total, mutu, dan keterangan berupa file pdf.	Sistem menerima request dan mencetak laporan Nolap, NIM, nama mahasiswa, mata kuliah, absen, tugas, UTS, UAS, total, mutu, dan keterangan berupa file pdf.
	2. Klik cetak laporan		

Berdasarkan tabel 2, dapat dilihat bahwa terdapat 35 *test case* yang menjadi hasil dari pengujian *software*, yaitu H-A, H-B, H-C, H-D, H-E dengan masing-masing ID mempunyai kode nomor 01 hingga 07. Selanjutnya hasil pengujian ini akan disimpulkan pada tabel berikutnya.

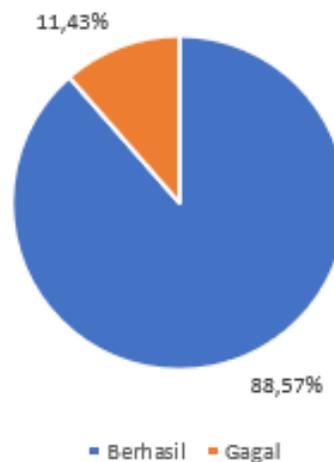
Tabel 2. Kesimpulan Hasil Pengujian Software

No.	ID	Kesimpulan
1	H-A01	Berhasil
2	H-A02	Berhasil
3	H - A03	Berhasil
4	H-A04	Berhasil
5	H-A05	Berhasil
6	H-A06	Gagal
7	H-A07	Berhasil
8	H-B01	Berhasil
9	H-B02	Berhasil
10	H-B03	Berhasil
11	H-B04	Berhasil
12	H-B05	Gagal
13	H-B06	Berhasil
14	H-B07	Berhasil
15	H-C01	Berhasil
16	H-C02	Gagal

No.	ID	Kesimpulan
17	H-C03	Berhasil
18	H-C04	Berhasil
19	H-C05	Gagal
20	H-C06	Berhasil
21	H-C07	Berhasil
22	H-D01	Berhasil
23	H-D02	Berhasil
24	H-D03	Berhasil
25	H-D04	Berhasil
26	H-D05	Berhasil
27	H-D06	Berhasil
28	H-D07	Berhasil
29	H-E01	Berhasil
30	H-E02	Berhasil
31	H-E03	Berhasil
32	H-E04	Berhasil
33	H-E05	Berhasil
34	H-E06	Berhasil
35	H-E07	Berhasil

Berdasarkan tabel 3, dapat dilihat bahwa dari 35 *test case* yang sebelumnya sudah diuji, didapatkan kesimpulan dengan total *test case* yang berhasil sebanyak 31 dan yang gagal sebanyak 4. Hasil kesimpulan akan dibuatkan visualisasi data berupa pie chart.

Data yang Dihasilkan



Gambar 7. Chart Hasil Pengujian Software

Pada gambar 7, merupakan pie chart dari hasil kesimpulan 35 *test case* dengan hasil sebanyak 11,43% gagal dan 88,57% berhasil. Hasil kesimpulan ini nantinya akan dilakukan evaluasi untuk mendapatkan perbaikan yang perlu dilakukan.

C. Evaluasi Hasil

Setelah selesai melakukan tes, langkah terakhir yang harus dilakukan yaitu mengevaluasi hasil. Berdasarkan hasil pengujian, menghasilkan hasil pengujian yang beragam. Dapat dipahami bahwa tes yang dilakukan telah lulus dan ada yang gagal. Hasil pengujian terdapat 35 *test case* dari aplikasi database Universitas Singaperbangsa Karawang dan 4 kasus uji yang diperoleh tidak sesuai dengan yang diharapkan. Meskipun ada 31 *test case* yang berhasil, ada hal-hal yang perlu ditingkatkan dari output yang dihasilkan oleh aplikasi.

Tabel 4. Evaluasi Hasil

No,	ID	Output yang dihasilkan	Rekomendasi
1	H-A H-B H-C H-D H-E	Tidak ada pesan berupa alert saat menambahkan, mengubah, dan menghapus data.	Perbaikan pesan: Sistem memberikan pesan alert seperti “Data gagal diinputkan, field harus diisi” ketika data tidak berhasil dimasukkan dan seterusnya.
2	H-A06 H-B05 H-C02 H-C05	Sistem menerima request walaupun data tidak diisi	Perbaikan sistem: Sistem memberikan alert berupa “Data belum ada yang diisi” untuk memberikan keterangan bahwa data belum sepenuhnya diisi sebelum menyimpan data.

Pengujian pada aplikasi saat melakukan input data, tidak memberikan alert yang membuat pengguna tidak mengetahui apakah input data berhasil dilakukan atau tidak. Akibatnya pada saat ada field yang kosong, semua data berhasil diinput. Seharusnya data tidak dapat diinput karena akan ada data yang tidak penting. Sebanyak 11,43% *test case* menghasilkan output gagal dan 88,57% berhasil. Meskipun hanya 11,43% gagal, perlu diketahui bahwa 11,43% hasil *test case* gagal tersebut merupakan uji pada fungsi fitur yang penting. Maka dari itu dengan dibuatnya evaluasi hasil, maka dapat dikatakan bahwa ada beberapa fungsi yang sesuai namun perlu adanya perbaikan agar aplikasi database ini dapat digunakan oleh staff perguruan tinggi dan terus berkembang menjadi lebih baik.

V. KESIMPULAN

Pengujian pada sebuah program penting untuk dilakukan dengan tujuan memeriksa apakah program tersebut berfungsi sesuai yang diinginkan atau tidak, sehingga pengujian sangat perlu dilakukan untuk mengurangi terjadinya kesalahan. Pada aplikasi database perguruan tinggi akan dilakukan testing dengan metode black box menggunakan teknik equivalence partition. Hasil pengujian memberikan hasil *mixed*. Hal tersebut dapat diartikan bahwa pengujian yang dilakukan ada yang berhasil dan ada yang gagal. Hasil pengujian ada 35 *test case* dari aplikasi database perguruan tinggi dan diperoleh 4 *test case* yang belum sesuai harapan. Meskipun ada 31 kasus uji yang berhasil, ada hal-hal yang perlu diperbaiki dari output yang dihasilkan oleh aplikasi. Sebanyak 11,43% *test case* belum sesuai harapan dan 88,57% berhasil. Meskipun hanya 11,43% gagal, perlu diketahui bahwa 11,43% hasil *test case* gagal tersebut merupakan uji pada fungsi fitur yang penting. Pada hasil yang sudah didapatkan, direkomendasikan untuk melakukan perbaikan agar aplikasi database perguruan tinggi dapat digunakan oleh staff perguruan tinggi dan terus berkembang lebih baik.

DAFTAR PUSTAKA

- [1] Ankita Sethi, “A REVIEW PAPER ON LEVELS, TYPES & TECHNIQUES IN SOFTWARE TESTING,” vol. 8, no. 7, pp. 269–271, 2017.

- [2] A. Z. Mubarok, Carudin, and A. Voutama, “Perancangan User Interface/User Experience Pada Aplikasi Baby Spa Berbasis Mobile Untuk User Customer Dan Terapis Menggunakan Metode User Centered Design,” *Al-Irsyad*, vol. 105, no. 2, p. 79, 2017, [Online]. Available: <https://core.ac.uk/download/pdf/322599509.pdf>.
- [3] S. Nidhra and J. Dondeti, “BLACK BOX AND WHITE BOX TESTING TECHNIQUES – A LITERATURE REVIEW,” vol. 2, no. 2, pp. 29–50, 2012.
- [4] N. Safitri and R. Pramudita, “Pengujian Black Box Menggunakan Metode Cause Effect Relationship Testing,” vol. 3, no. 1, pp. 101–110, 2018.
- [5] E. Novalia and A. Voutama, “Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah,” vol. 11, no. 11, pp. 23–34, 2022.
- [6] A. Voutama and E. Novalia, “Perancangan Aplikasi M-Magazine Berbasis Android Sebagai Sarana Mading Sekolah Menengah Atas,” vol. 15, no. 1, pp. 104–115.
- [7] A. A. Arwaz, K. Putra, R. Putra, and T. Kusumawijaya, “Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions,” vol. 2, no. 4, pp. 130–134, 2019.
- [8] S. Sistem, I. Universitas, N. Pgri, K. Kediri, and I. E-mail, “Perancangan Active Database System pada Sistem Informasi Pelayanan Harga Pasar,” vol. 1, no. 1, pp. 35–43, 2017.
- [9] C. Thomas and C. Begg, *Database Systems A Practical Approach to Design, Implementation, and Management*. 2010.
- [10] V. A. Amei *et al.*, “Pengujian Website ACC . CO . ID Revamp Menggunakan Metode Black Box Testing,” pp. 39–46, 2021.
- [11] R. S. Pressman, *Software Engineering : A Practitioner’s Approach (Seventh Edition)*. 2010.
- [12] S. Roohullah Jan, S. Tauhid Ullah Shah, Z. Ullah Johar, Y. Shah, and F. Khan, “An Innovative Approach to Investigate Various Software Testing Techniques and Strategies,” *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 682–689, 2016.
- [13] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, “Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 155, 2019, doi: 10.32493/jtsi.v2i4.3771.
- [14] D. Widhyaestoeti *et al.*, “BLACK BOX TESTING EQUIVALENCE PARTITIONS UNTUK PENGUJIAN FRONT-END PADA SISTEM AKADEMIK SITODA,” vol. 7, no. 3, 2021.
- [15] Wamiliana, I. Adi, E. Heppyda, and W. Wardhana, “Pembangunan Tools Pembelajaran Pemrograman Java Melalui Live CD / Live USB,” *J. Komputasi*, vol. 3, no. 1, pp. 95–105, 2015, [Online]. Available: <http://jurnal.fmipa.unila.ac.id/index.php/komputasi>.
- [16] H. Maulana, “Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source,” *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 1, pp. 32–37, 2016, doi: 10.30743/infotekjar.v1i1.37.
- [17] R. Pramudita, “Pengujian Black Box pada Aplikasi Ecampus Menggunakan Metode Equivalence Partitioning,” vol. 4, no. 2, pp. 193–202, 2020.
- [18] A. Amalia, S. W. Putri Hamidah, and T. Kristanto, “Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web,” *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 269–274, 2021, doi: 10.47065/bits.v3i3.1062.