

**TICKETING SYSTEM UNTUK PELAYANAN MAHASISWA
INTERNASIONAL MENGGUNAKAN DJANGO**



**Disusun sebagai salah satu syarat memperoleh Gelar Strata I
pada Program Studi Informatika Fakultas Komunikasi dan Informatika**

Oleh:
ALI MUSTHOFA
L200170165

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

2022

HALAMAN PERSETUJUAN

**TICKETING SYSTEM UNTUK PELAYANAN MAHASISWA INTERNASIONAL
MENGUNAKAN DJANGO**

PUBLIKASI ILMIAH

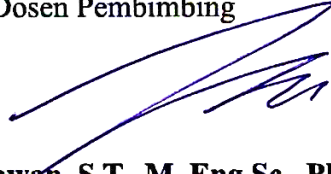
Oleh

ALI MUSTHOFA

L200170165

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing



Fajar Suryawan, S.T., M. Eng.Sc., Ph.D.

NIK. 924

HALAMAN PERSETUJUAN

TICKETING SYSTEM UNTUK PELAYANAN MAHASISWA INTERNASIONAL MENGUNAKAN DJANGO

Oleh :

ALI MUSTHOFA

L200170165

**Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari senin, 13 Februari 2023
dan dinyatakan telah memenuhi syarat**

Dewan Penguji :

- 1. Fajar Suryawan, S.T., M.Eng.Sc., Ph.D** (.....) (Ketua Dewan Penguji)
- 2. Dedi Gunawan, S.T., M.Sc., Ph.D.** (.....) (Anggota I Dewan Penguji)
- 3. Nurgiyatna, S.T., M.Sc., Ph.D.** (.....) (Anggota II Dewan Penguji)



**Dekan
Fakultas Komunikasi dan Informatika**

Nurgiyatna, S.T., M.Sc., Ph.D.
NIK. 881



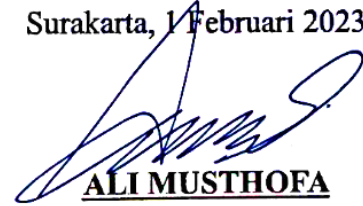
**Ketua
Program Studi Informatika**
Dedi Gunawan, S.T., M.Sc., Ph.D.
NIK. 1305

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam publikasi ilmiah ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan diterbitkan dalam daftar pustaka

Apabila kelak terbukti ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya

Surakarta, 1 Februari 2023



ALI MUSTHOFA

L200170165

TICKETING SYSTEM UNTUK PELAYANAN MAHASISWA INTERNASIONAL MENGGUNAKAN DJANGO

Abstrak

Biro Kerjasama dan Urusan Internasional merupakan salah satu biro yang dimiliki oleh Universitas Muhammadiyah Surakarta yang bertanggung jawab atas berjalannya masa studi bagi mahasiswa internasional. Kerap kali mahasiswa internasional menghadapi masalah selama studi mereka dan karenanya membutuhkan bantuan untuk menyelesaikan masalah mereka. Namun, dokumentasi aduan masih relatif tidak efektif dalam menyelesaikan permasalahan yang mereka hadapi, sehingga kinerja petugas yang bertugas dapat terpengaruh. *Ticketing System* akan berguna dalam mendokumentasikan setiap aduan keluhan yang dimiliki mahasiswa internasional terhadap permasalahan yang terjadi selama menjalani masa studi. Metode pengembangan sistem informasi *ticketing system* ini menggunakan metode *Waterfall*, serta tahapan metode *Waterfall* adalah analisis, desain, pengkodean, pengujian, dan pemeliharaan. Tools yang digunakan Django sebagai *framework full-stack* dengan bahasa pemrograman *Python*, *PostgreSQL* sebagai database. Hasil penelitian ini menghasilkan sistem informasi *Ticketing System* yang dapat memberikan kemudahan terhadap staf dalam melakukan pencatatan keluhan permasalahan, serta mahasiswa juga dapat melihat status informasi penyelesaian masalah yang pernah diajukan.

Kata kunci : *Ticketing System*, Django, Mahasiswa Internasional.

Abstract

The Bureau of Cooperation and International Affairs is one of the bureaus owned by the Muhammadiyah University of Surakarta which is responsible for the course of the study period for international students. Many times international students face problems during their studies and thus need help to solve their problems. However, complaint documentation is still relatively ineffective in resolving the problems they face, so that the performance of the officers on duty can be affected. The ticketing system will be useful in documenting any complaints that international students have regarding problems that occur during their study period. The method of developing this ticketing system information system uses the Waterfall method, and the stages of the Waterfall method are analysis, design, coding, testing, and maintenance. Tools used by Django as a full-stack framework with the Python programming language, PostgreSQL as a database. The results of this study resulted in a Ticketing System information system that can provide convenience to staff in recording problem complaints, and students can also see the status of problem solving information that has been submitted.

Keywords: Ticketing System, Django, International Student.

1. PENDAHULUAN

Seiring waktu, perkembangan teknologi yang pesat di bidang komputer, elektronik, komunikasi, dan permesinan telah menyebabkan berbagai aplikasi kompleks dan cerdas memainkan peran penting dalam kehidupan manusia. Dalam suatu perusahaan peningkatan

kualitas usaha tentunya memerlukan pengelolaan yang baik agar pekerjaan berjalan efektif dan efisien. Tentunya selain untuk memaksimalkan proses pengelolaan perusahaan, proses monitoring juga tidak kalah pentingnya untuk meningkatkan kualitas pekerjaan dengan tujuan peningkatan efektivitas pekerjaan. Organisasi semakin melihat ke arah penerapan teknologi tidak hanya untuk mendukung operasi bisnis yang ada tetapi juga untuk menciptakan peluang baru yang memberi mereka sumber keunggulan kompetitif (Ward & Peppard, 2002).

Teknologi telah merambah banyak bidang, termasuk pendidikan. Pihak-pihak dalam dunia pendidikan harus mampu mengimbangi, memantau dan beradaptasi dengan perkembangan teknologi. Dengan kemajuan teknologi, setiap pendokumentasian atau pencatatan (komunikasi mahasiswa dengan kampus) dari permasalahan yang dihadapi oleh mahasiswa memperkenalkan sistem pencatatan dan pengelolaan yang sistematis, karena pengelolaan data secara manual saat ini dinilai kurang efektif dan efisien. Oleh karena itu diperlukan sistem informasi *helpdesk* agar sistem dapat bekerja dengan baik dan memiliki rencana tindakan yang cepat dalam situasi bermasalah.

Menurut Fred, *helpdesk* dapat berfungsi sebagai *single point of contacts* ketika pelanggan membutuhkan bantuan teknis serta dapat menjadi fasilitas komunikasi antara pelanggan dengan tim pendukung dalam sebuah perusahaan (Beisse, 2015). Sistem *helpdesk* telah menjadi kebutuhan vital bagi institusi besar yang sangat bergantung pada layanan dan sumber daya. Ini karena bertindak sebagai pusat informasi antara staf dan pengguna terkait layanan yang diminta dan masalah yang dilaporkan (Al-Hawari & Barham, 2021). *Helpdesk* bertujuan untuk memfasilitasi semua penggunaan perangkat IT dalam lingkup perusahaan dan instansi yang melaporkan setiap masalah yang dihadapi (Qoyyimah et al., 2012).

Penelitian sebelumnya dibuat dalam studi tahun 2020 sebelumnya oleh Husen, berfungsi sebagai sarana untuk melaporkan masalah perangkat keras atau perangkat lunak apa pun yang terjadi di perusahaan yang disurvei. Aplikasi bersifat online dan pelaporan penggunaan aplikasi bertujuan untuk memudahkan pelaporan masalah yang diidentifikasi oleh karyawan dan dilakukan sesuai dengan SLA (*Service Level Agreement*) yang ditetapkan oleh perusahaan. Aplikasi ini menggunakan bahasa pemrograman *PHP* dan *CodeIgniter* sebagai *backend* aplikasi. Metode pengembangan aplikasi menggunakan metode *waterfall* untuk mengembangkan system (Husen, 2020). Untuk topik penelitian yang sama, Qoyyimah membuat sistem ticketing help desk (studi kasus: PT Primus Indojoya) membantu perusahaan mengelola laporan. Dengan bantuan sistem ini, help desk dapat dikomputerisasi dengan bantuan sistem sehingga informasi yang terkait dengan *helpdesk* dapat disimpan secara terpusat di database dan juga laporan dapat dibuat secara otomatis di dalam sistem. Dalam pengembangan

sistem menggunakan metode *waterfall* dengan pemodelan *Unified Modelling Language* (UML). Penggunaan bahasa pemrograman *PHP* dan *MySQL* sebagai database (Qoyyimah et al., 2012).

Selain itu, Rizki Mudhar menyampaikan topik penelitian yang sama dalam penelitiannya yang berjudul “Pengembangan Sistem Informasi *Helpdesk Ticketing* Menggunakan *Django Framework*”. Membuat sistem informasi *helpdesk* yang memudahkan karyawan untuk membantu memecahkan masalah yang muncul dan dapat memberikan solusi yang tepat dan cepat. Bahasa pemrogramannya adalah *Django* yang menggunakan *framework Django*, dan databasenya adalah *MySQL*. Dalam metode pengembangan sistem, peneliti menggunakan pemodelan *Rapid Application Development* (RAD) dan *Unified Modeling Language* (UML) (Mudhar, 2015).

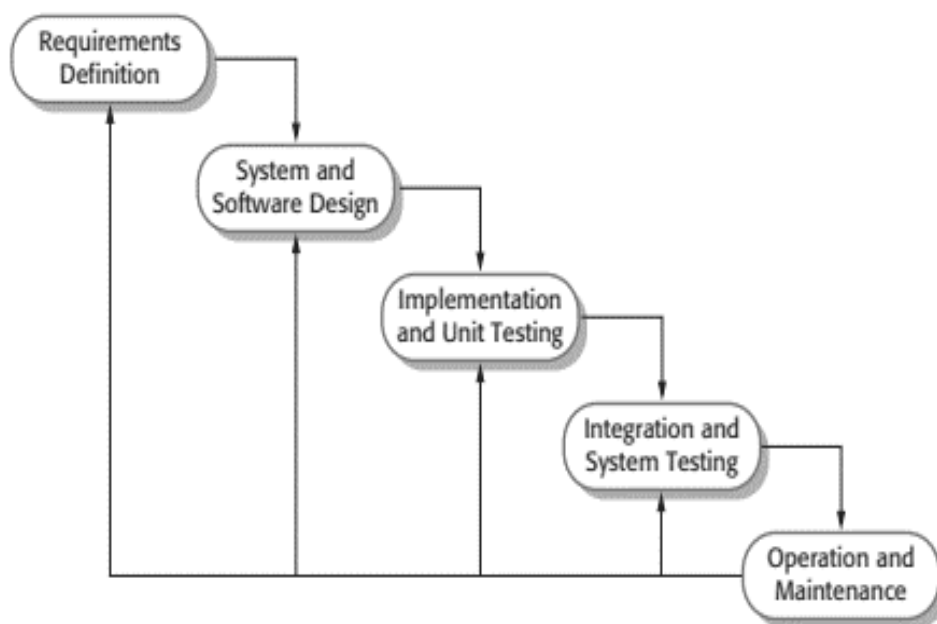
Melihat dari beberapa penelitian sebelumnya, dapat dikatakan bahwa dibangunnya *ticketing system helpdesk* dapat mempermudah pekerjaan staf dalam melakukan pencatatan, dokumentasi, serta tanggap dalam menangani masalah dengan mudah karena terpusat di sistem yang dibangun. Dengan semakin banyaknya pengaduan yang terjadi setiap hari, penggunaan email dan telepon dinilai kurang efektif dan efisien dalam menangani pengaduan tersebut karena semakin sulitnya mengontrol pengaduan dari pengguna (Rachmawati & Suhendra, 2018). Tentu saja, ini dapat memberikan dampak yang kurang baik terhadap kinerja staf dalam membantu mahasiswa internasional untuk menyelesaikan masalah mereka. Dampaknya dapat berupa: masalah mungkin direkam berulang kali, ada masalah yang tidak dicatat, tinjauan masalah kurang tepat karena dilakukan secara manual dan waktu pemrosesan terhambat karena kebutuhan karyawan harus menghubungi staf terlebih dahulu dan dapat meminta penjelasan kepada staf masalah apa yang terjadi, sehingga membutuhkan waktu yang cukup lama.

Pada penelitian ini, bahasa pemrograman yang digunakan yaitu *Python* dengan *framework Django* seperti penelitian Rizki Mudhar. Bedanya pada penelitian ini digunakan metode *waterfall*. Alasan penggunaan metode itu sendiri adalah metode tersebut terasa lebih konsisten dan fokus serta proses pengembangannya tidak tumpang tindih. Ketika pemberitahuan diterima, secara otomatis diteruskan ke *front desk*, yang akan memproses masalah tersebut sesuai laporan. *Ticketing system* merupakan sistem informasi yang sebagai jembatan komunikasi antara client dengan perusahaan. *Ticketing system* dinilai mampu secara teknis menyederhanakan pekerjaan staf karena dapat mendistribusikan pekerjaan antar staf secara merata, dapat memfasilitasi pelaporan pertanyaan, layanan, dukungan teknis atau beberapa keluhan terkait layanan sehingga teknisi dapat segera mengidentifikasi mereka dengan benar.

Diharapkan dengan adanya sistem informasi *ticketing system* untuk pelayanan mahasiswa internasional ini dapat membantu staf dalam mengumpulkan, menyimpan dan mengolah informasi tentang mahasiswa internasional terkait permasalahan yang mereka hadapi. Hal ini juga dapat mempermudah pekerjaan *front desk* karena dapat mendistribusikan pekerjaan secara merata di antara staf. Selain itu, sistem informasi ini dapat memudahkan pelaporan masalah yang dihadapi mahasiswa, memungkinkan staf untuk segera mengidentifikasi masalah dan membantu menyelesaikannya.

2. METODE

Pengembangan sistem informasi *ticketing system* untuk pelayanan mahasiswa internasional menggunakan *Software Development Life Cycle* (SDLC) dengan model *waterfall*. Metode ini merupakan proses sekuensial linier dimana setiap langkah hanya dimulai setelah langkah sebelumnya diselesaikan. Sedangkan teknik pengumpulan data mengandalkan wawancara dan kajian pustaka untuk mendapatkan informasi yang digunakan dalam analisis kebutuhan dan perancangan sistem. Gambar 1 visualisasi dari tahapan metode waterfall.



Gambar 1. Metode *Waterfall* (Sommerville, 2011)

2.1. Requirement Definiton

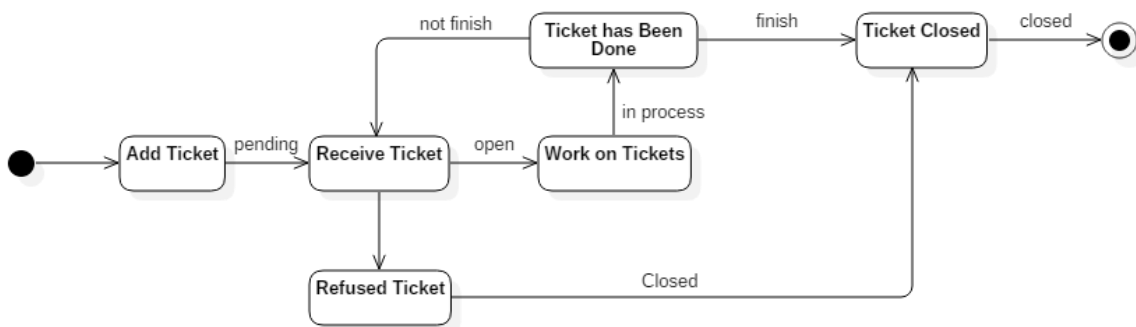
Langkah awal sebelum pengembangan sistem, fase ini mencari informasi model aplikasi, spesifikasi yang dibutuhkan untuk pengembangan sistem informasi dan analisis seperti kebutuhan fungsional dan non fungsional. *Tools* yang digunakan dalam pengembangan penelitian ini menggunakan *framework Django* dengan bahasa pemrograman *Python* untuk membuat website.

2.2. System and software design

Setelah menganalisis kebutuhan sistem, langkah selanjutnya adalah merancang sistem. Fase ini merupakan gambaran dari proses pengembangan sistem yang meliputi *state diagram*, *use case diagram*, *activity diagram* dan *entity relationship diagram* (ERD) dari sistem yang akan dikembangkan.

2.2.1. Statechart Diagram

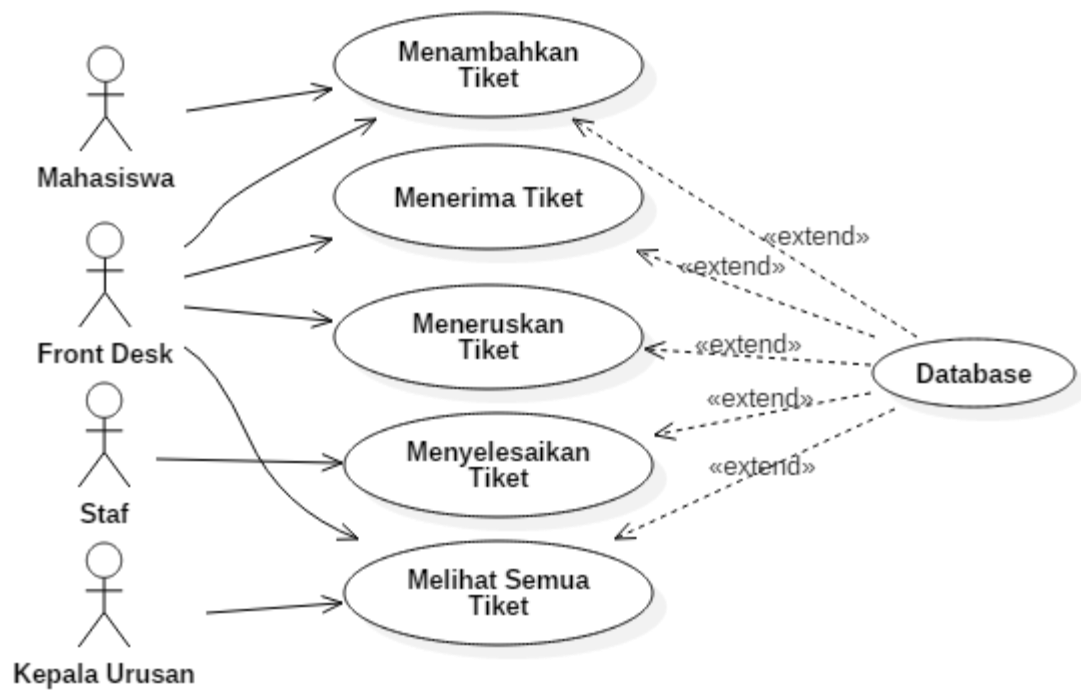
State diagram memodelkan siklus hidup suatu objek. Ini menjelaskan *state* berbeda yang dapat dimiliki suatu objek, peristiwa yang menyebabkan objek berubah *state* dari waktu ke waktu, dan aturan yang mengatur transisi objek antar *state*. Dengan kata lain, ini menentukan dari *state* mana suatu objek diizinkan untuk bertransisi ke *state* yang lain dan dalam kondisi apa (Whitten & Bentley, 2007). Gambar 2 visualisasi dari *statechart diagram* pada sistem ini.



Gambar 2. Statechart Diagram Proses Penyelesaian Tiket

2.2.2. Use Case Diagram

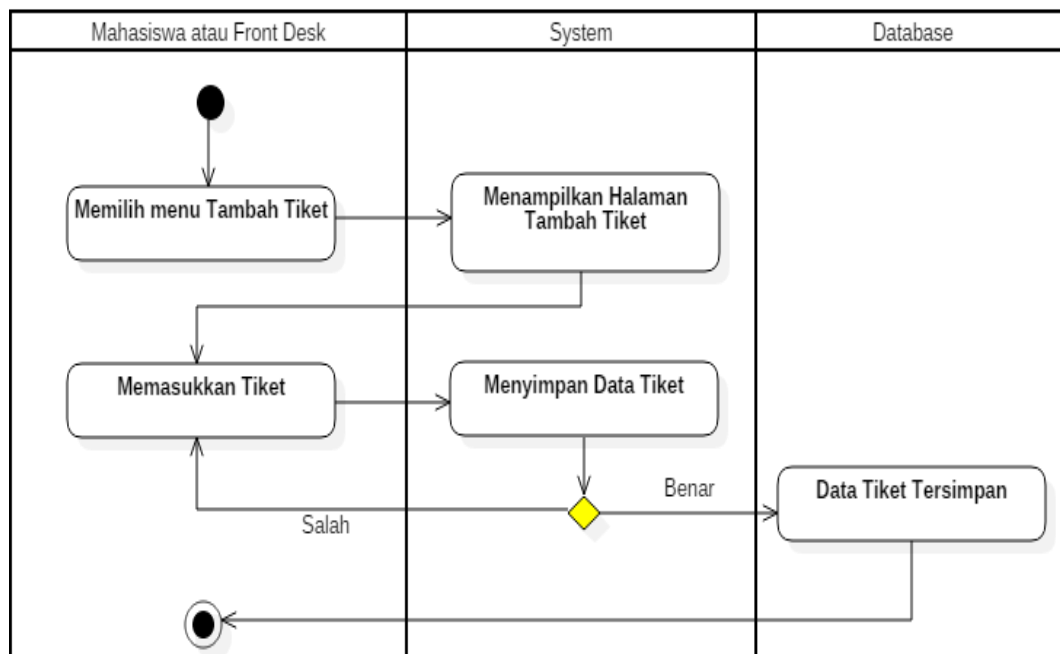
Salah satu diagram yang paling penting untuk mengilustrasikan persyaratan sistem adalah diagram *Use Case* (UC), yang dengan jelas menjelaskan konteks interaksi antara aktor dan sistem. Use case diagram juga dapat digunakan untuk mengetahui fungsi mana saja yang dapat digunakan pada sistem. Terdapat 4 aktor dalam sistem ini yaitu kepala urusan, staf, *front desk*, dan mahasiswa. Gambar 3 visualisasi dari *Use Case Diagram* pada sistem ini.



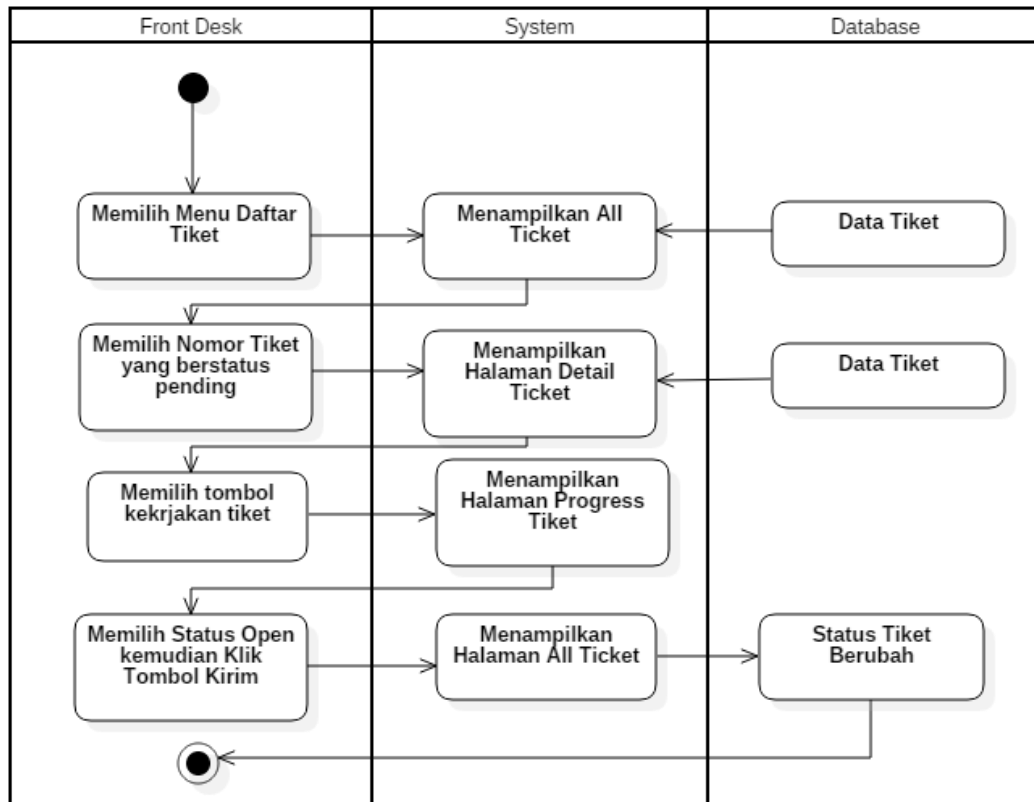
Gambar 3. Visualisasi Use Case Diagram

2.2.3. Activity Diagram

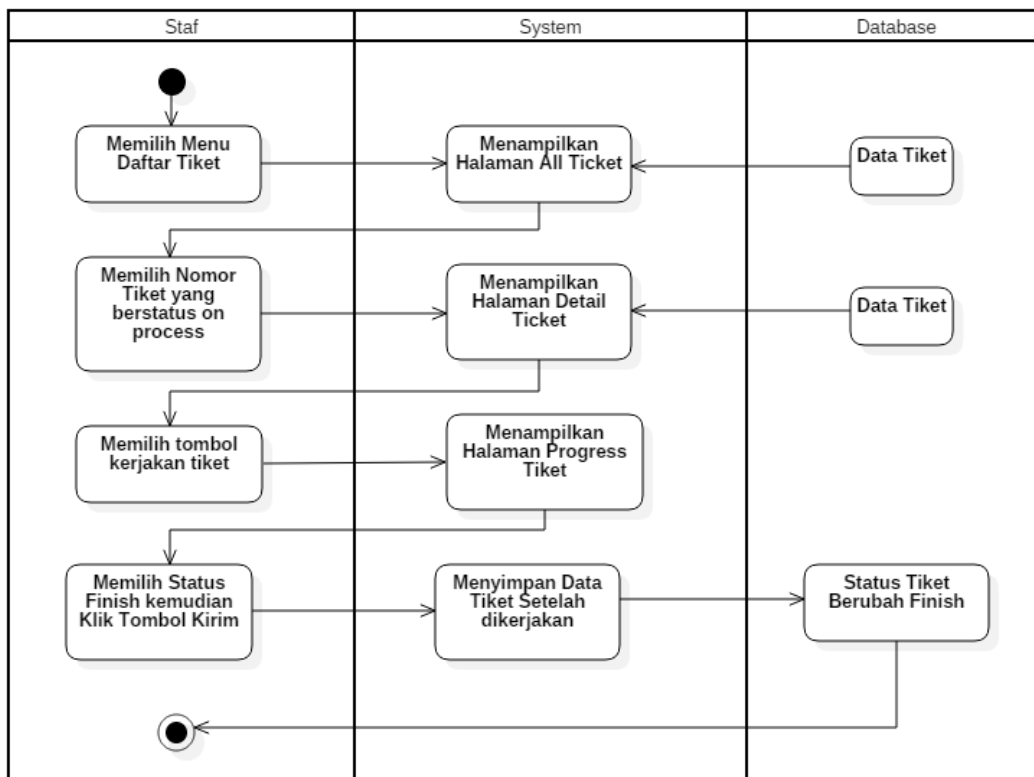
Diagram aktivitas menggambarkan aliran aktivitas yang berbeda dari sistem yang direncanakan, bagaimana setiap aliran dimulai, keputusan apa yang dapat dibuat dan bagaimana akhirnya. Diagram aktivitas untuk penelitian ini meliputi *activity diagram* Menambahkan Tiket, *activity diagram* Meneruskan Tiket, *activity diagram* Menyelesaikan Tiket, dan *activity diagram* Melihat Tiket. Gambar 4-7 merupakan visualisasi dari *activity diagram* dari masing-masing aktor.



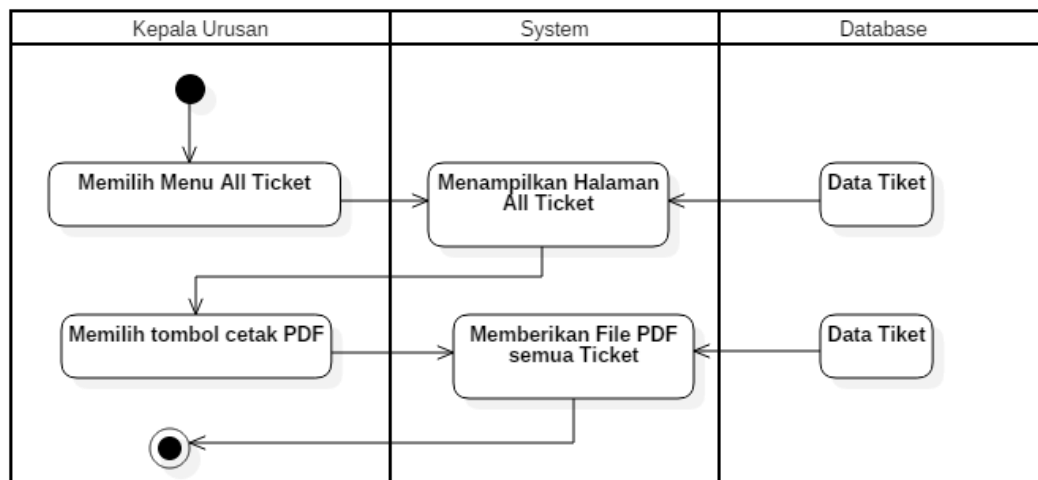
Gambar 4. Visualisasi Activity Diagram Menambahkan Tiket



Gambar 5. Visualisasi Activity Diagram Meneruskan Tiket



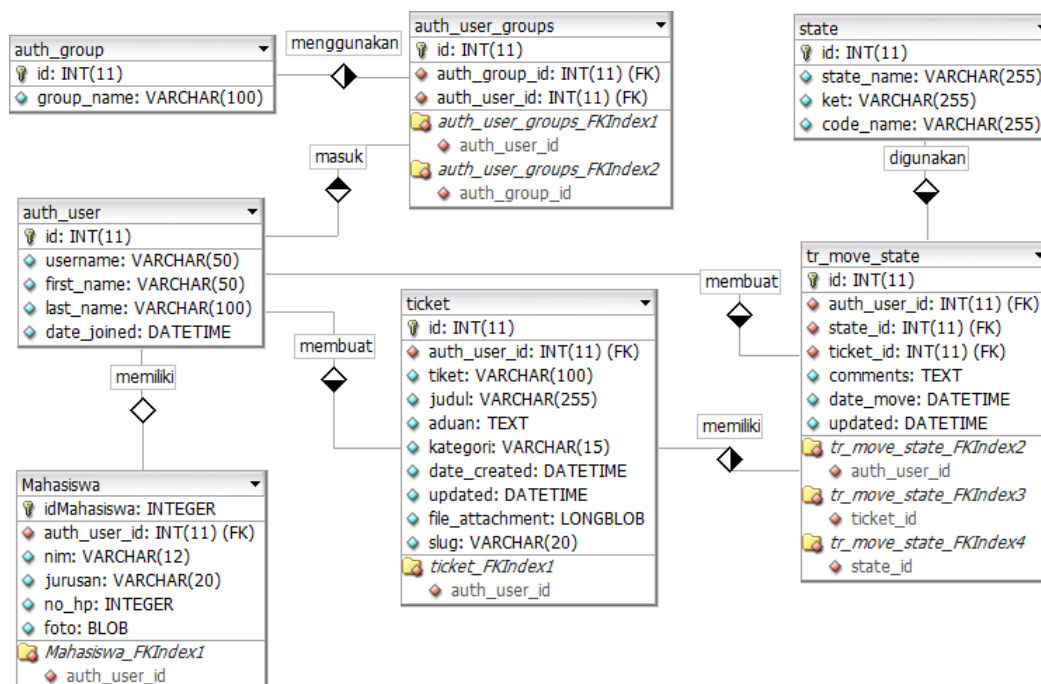
Gambar 6. Visualisasi Activity Diagram Menyelesaikan Tiket



Gambar 7. Visualisasi Activity melihat laporan

2.2.4. ER Diagram

Diagram relasi adalah diagram yang digunakan dalam desain basis data yang menunjukkan hubungan antara satu tabel dengan tabel lainnya. Gambar 8 visualisasi hubungan antar tabel.



Gambar 8. Entity-relationship Diagram

2.3. Implementation and unit testing

Langkah selanjutnya dari metode *waterfall* adalah implementasi kode program menggunakan berbagai *tools* dan bahasa pemrograman sesuai dengan kebutuhan yang telah disiapkan sebelumnya. Jadi pada tahap implementasi ini lebih difokuskan pada masalah teknis dimana hasil dari desain perangkat lunak diterjemahkan ke dalam bahasa pemrograman yang

membentuk sistem. Pengembangan *ticketing system* menggunakan framework Django dengan bahasa pemrograman *python*, serta database *PostgreSQL*.

2.4. Integration and unit testing

Pengujian sistem diperlukan untuk menentukan apakah sistem berfungsi. Perannya sangat penting karena pengujian sistem merupakan salah satu fase waterfall dan berkaitan dengan keandalan sistem yang dibangun. Dalam pengembangan sistem informasi ini, metode *black box* dan *System Usability Scale* (SUS) digunakan sebagai metode pengujian. Pengujian *black box* memainkan peran penting dalam pengujian perangkat lunak, mereka membantu memvalidasi fungsionalitas sistem secara keseluruhan. Pengujian *black box* dilakukan berdasarkan kebutuhan pelanggan, sehingga kebutuhan yang tidak lengkap atau tidak terduga dapat dengan mudah diidentifikasi dan ditangani kemudian (Nidhra & Dondeti, 2012).

2.5. Operation and maintenance

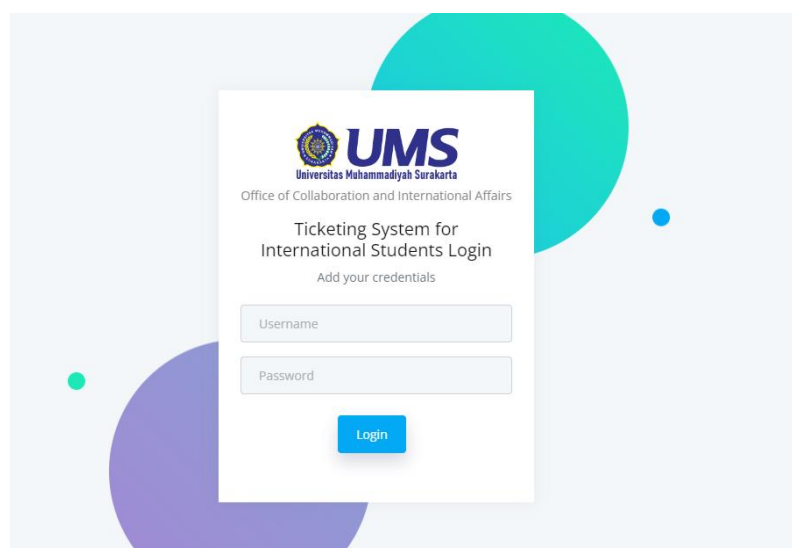
Pemeliharaan meliputi perbaikan bug yang tidak ditemukan pada tahap sebelumnya, peningkatan implementasi unit sistem dan peningkatan layanan sistem sebagai persyaratan baru.

3. HASIL DAN PEMBAHASAN

3.1. Hasil

3.1.1. Tampilan Awal Halaman Website

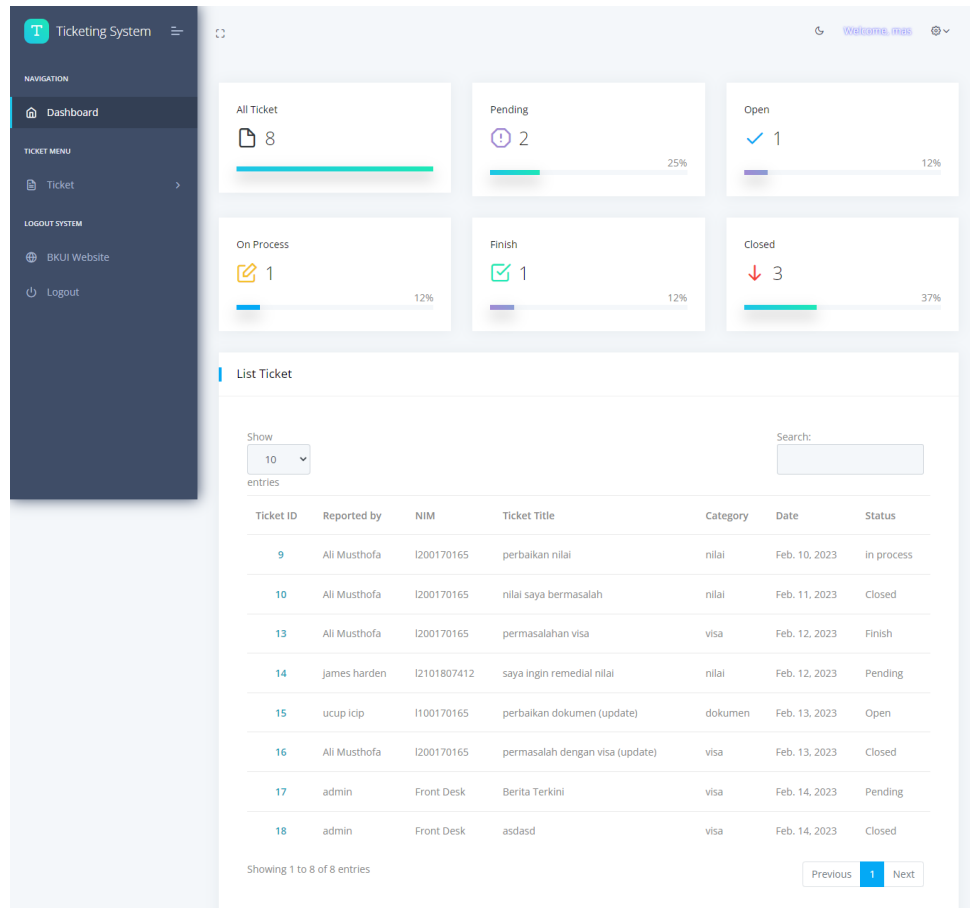
Pada *ticketing system* ini menampilkan halaman awal berupa *login form*, dikarenakan sistem ini dirancang untuk membuat sebuah laporan ketika terjadi permasalahan yang terjadi dengan mahasiswa internasional selama menjalani perkuliahan di Universitas Muhammadiyah Surakarta, Gambar 9 merupakan visualisasi tampilan awal halaman website.



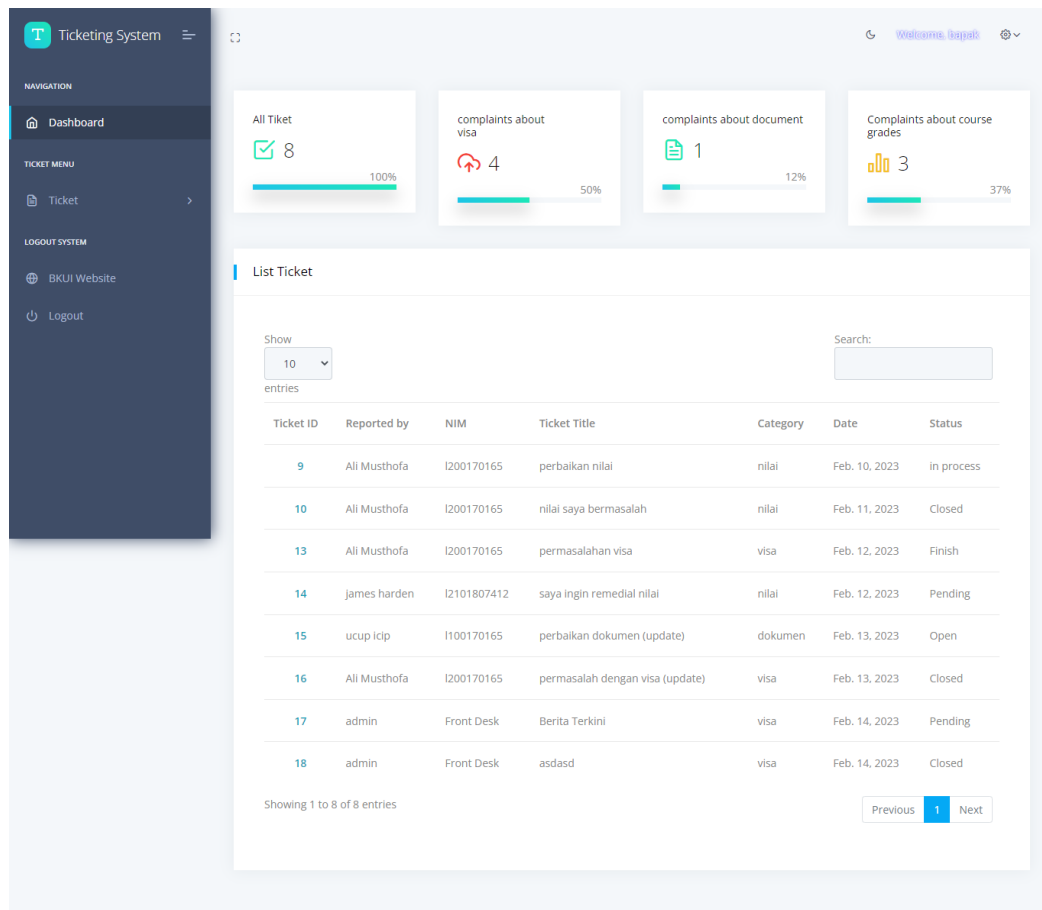
Gambar 9. Visualisasi Halaman Login

3.1.2. Tampilan Halaman *Dashboard* atau *Home*

Halaman *dashboard* menampilkan berupa jumlah data yang sudah masuk, serta memberikan tampilan list tiket. Untuk aktor kaur akan memiliki tampilan yang berbeda dengan dengan aktor lainnya. Gambar 10-11 merupakan visualisasi tampilan halaman *dashboard*.



Gambar 10. Tampilan Halaman Dashboard *Front Desk*



Gambar 11. Tampilan Halaman Dashboard Kaur

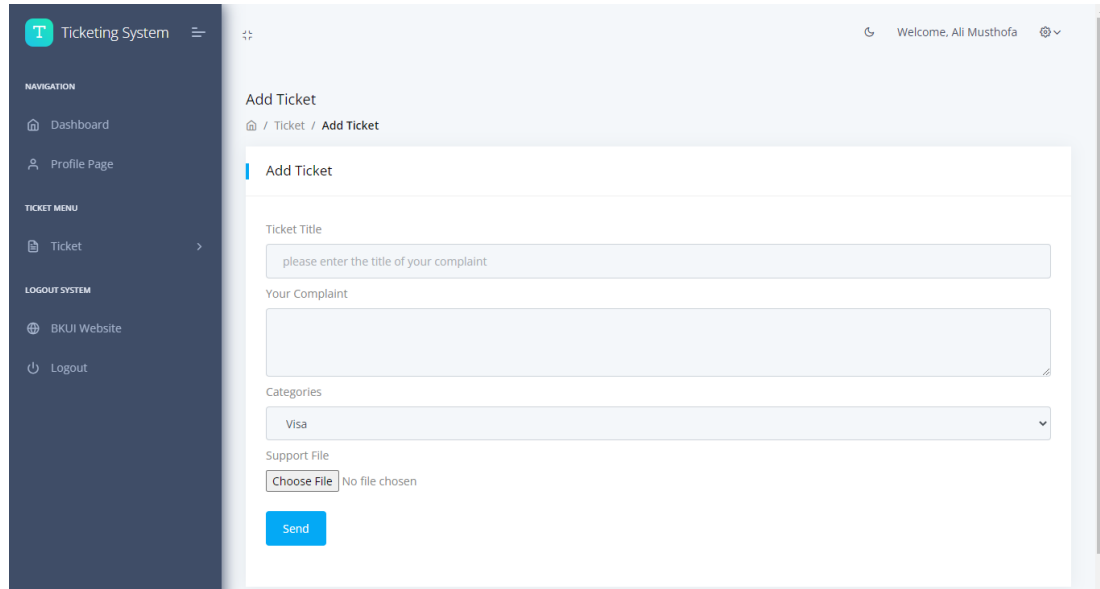
3.1.3. Tampilan Halaman Profil Mahasiswa

Menampilkan data diri mahasiswa, pada halaman ini mahasiswa hanya dapat melakukan update pada nomer handphone dan juga mengganti foto mereka, Gambar 12 merupakan visualisasi tampilan halaman profil mahasiswa.

Gambar 12. Tampilan Halaman Profil Mahasiswa

3.1.4. Tampilan Halaman *Add Ticket*

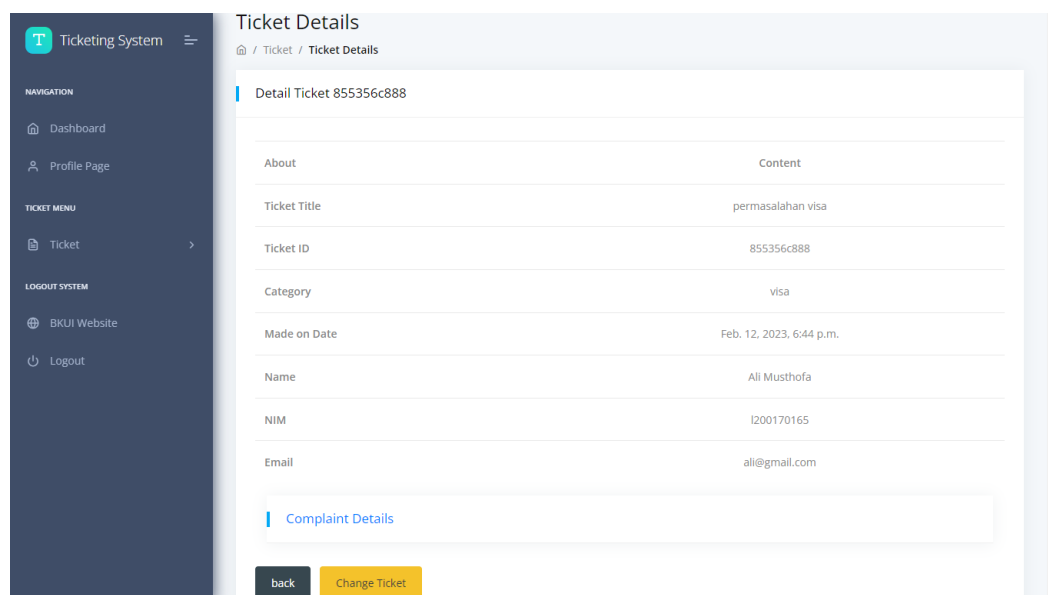
Di halaman ini dapat menambahkan tiket baru untuk mahasiswa dan juga untuk *front desk* dengan mengisi jenis aduan, aduan, kategori aduan, dan file dukungan. ditampilkan ditampilkan pada Gambar 13 merupakan visualisasi tampilan halaman add tiket.



Gambar 13. Tampilan Halaman Menambahkan Tiket

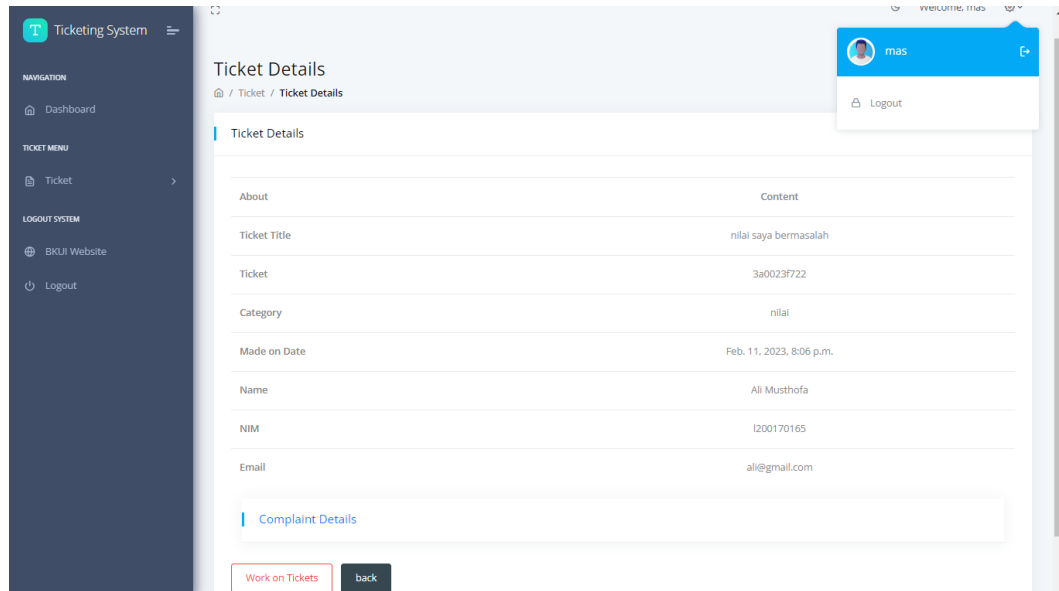
3.1.5. Tampilan Halaman *Tiket Detail*

Setelah mahasiswa atau *front desk* melakukan input tiket pada menu add tiket, akan ditampilkan *view ticket detail* untuk melihat hasil tiket yang sudah di inputkan. Ketika ada kesalahan dalam melakukan input, maka mahasiswa dapat *edit* data tersebut dengan klik button update ditampilkan pada Gambar 14 merupakan visualisasi tampilan halaman tiket detail.



Gambar 14. Tampilan Halaman *Detail Ticket* Mahasiswa

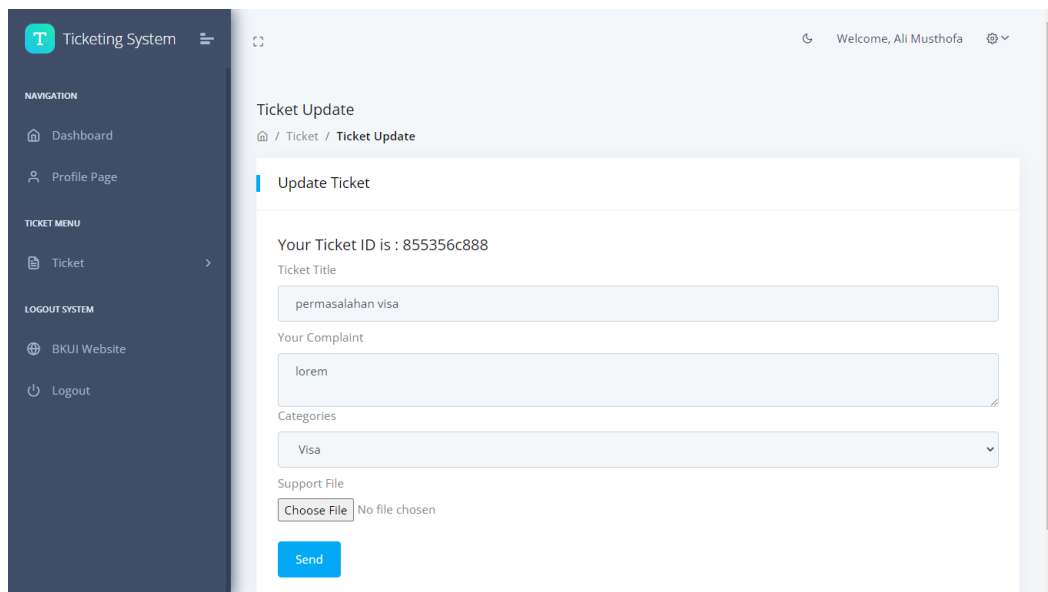
Selanjutnya merupakan menu detail tiket untuk Kaur, Staf, serta *Front Desk* ada tombol kerjakan tiket, dimana ketika tombol ini akan meneruskan ke halaman membuat status atau melakukan perubahan status terhadap tiket tersebut ditampilkan pada Gambar 15 merupakan visualisasi tampilan menu detail tiket.



Gambar 15. Tampilan Halaman Detail Front Desk

3.1.6. Tampilan Halaman *Change Ticket*

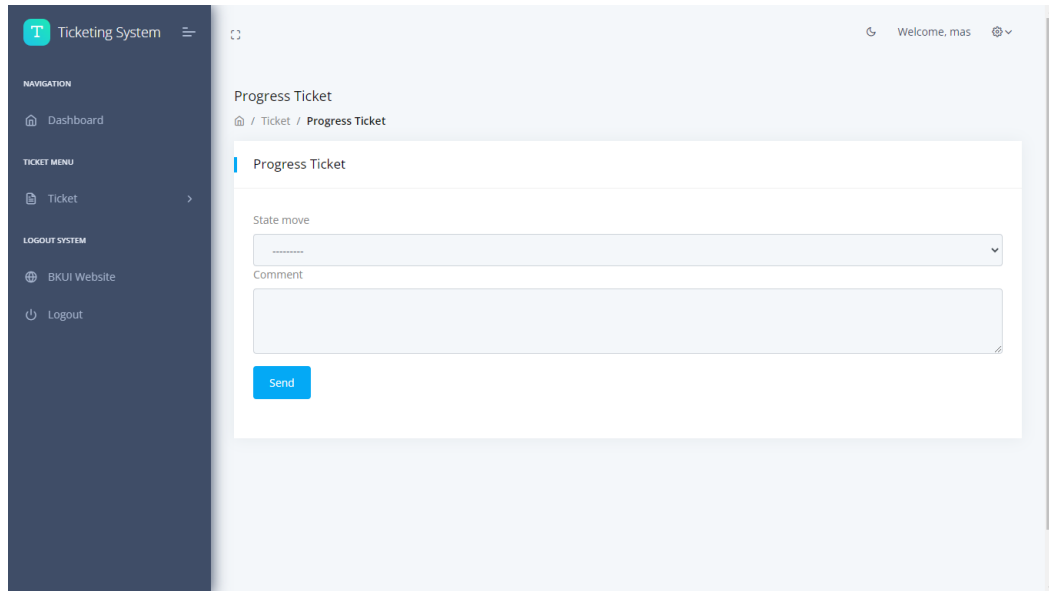
Mahasiswa yang melakukan input data dapat melakukan perubahan data di halaman ini, Mahasiswa dapat memperbaiki ketika ada kesalahan dalam melakukan input tiket ditampilkan pada Gambar 16 merupakan visualisasi tampilan halaman *change ticket*.



Gambar 16. Tampilan Halaman *Change Ticket*

3.1.7. Tampilan Halaman *Progress Ticket*

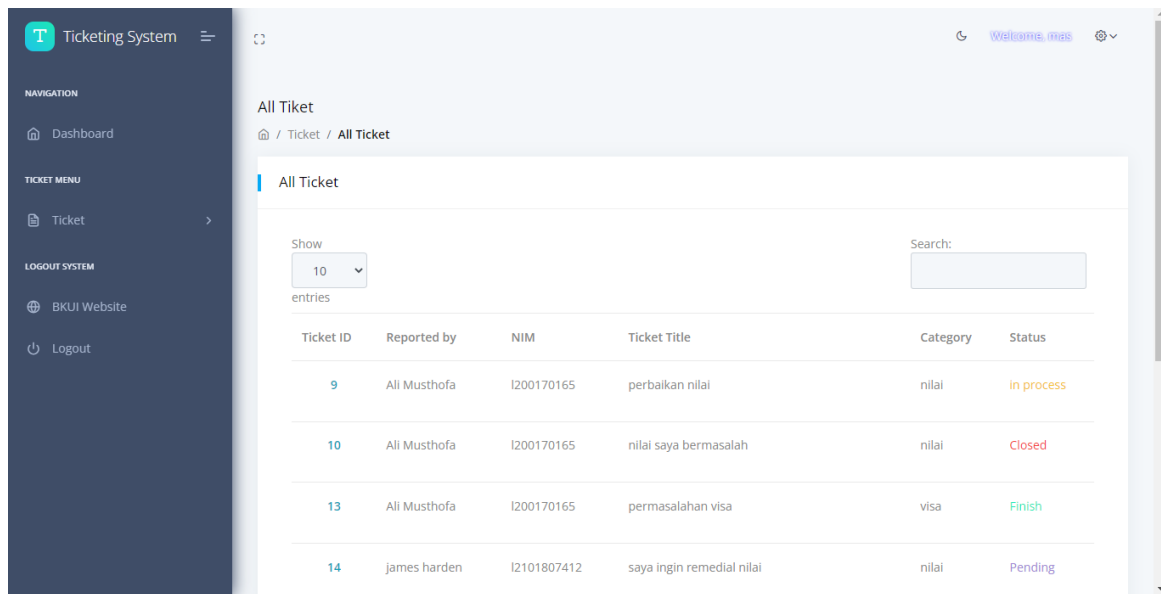
Halaman ini berfungsi untuk merubah status dari tiket, dengan memilih status yang sesuai dengan keadaan tiket sekarang, serta dapat memberikan komentar terhadap tiket yang berubah statusnya, untuk statusnya terdapat 5 kondisi *state* yaitu (*pending, open, process, finish, closed*), Gambar 17 merupakan visualisasi tampilan halaman *progress ticket*.



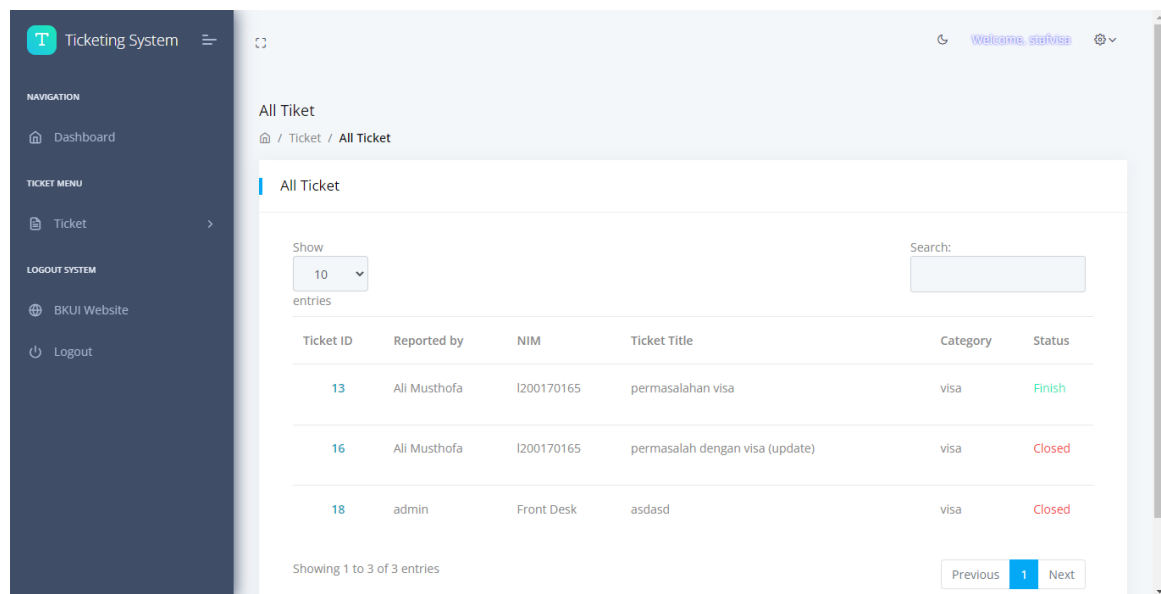
Gambar 17. Tampilan Halaman Menambahkan Status Tiket

3.1.8. Tampilan Halaman *View All Ticket*

View tiket ini berisi tiket yang telah di input oleh mahasiswa, sesuai dengan masing-masing mahasiswa yang melakukan input data. Jadi mahasiswa satu tidak dapat melihat tiket yang diinput oleh mahasiswa lainnya. Tetapi ketika *front desk*, staf, serta kaur masuk ke halaman *view* ini dapat melihat semua tiket yang telah diinput oleh mahasiswa dan *front desk*, terkhusus untuk staf tidak dapat melihat dengan status *pending*, ditampilkan pada Gambar 18-20 merupakan visualisasi tampilan halaman *view all ticket*.

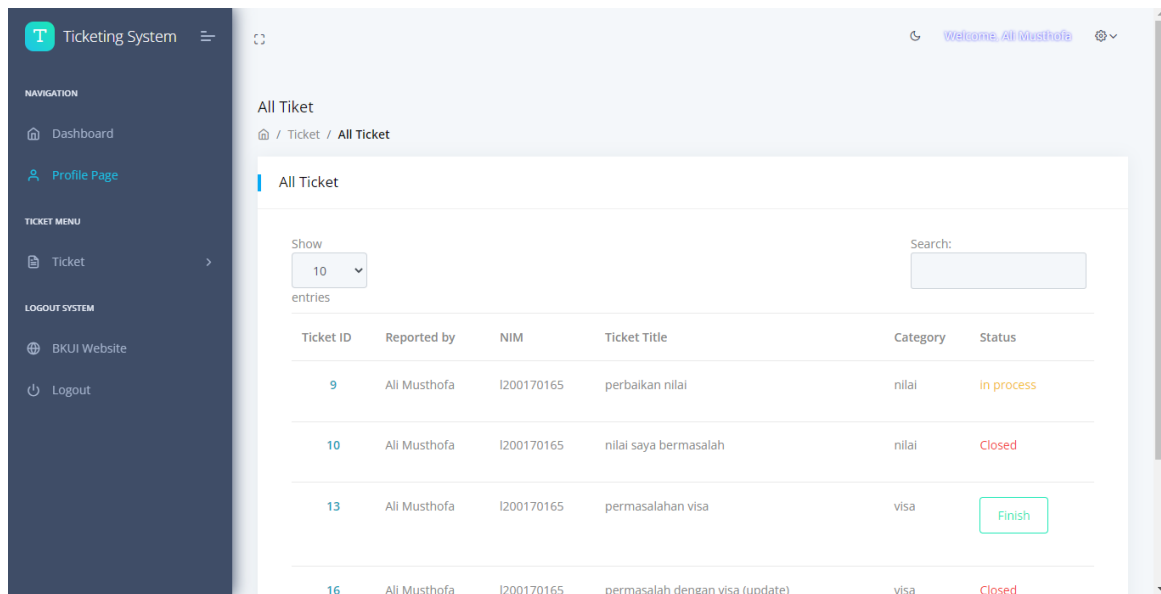


Gambar 18. Tampilan Halaman Semua Tiket *Front Desk*



Gambar 19. Tampilan Halaman Semua Tiket Staf

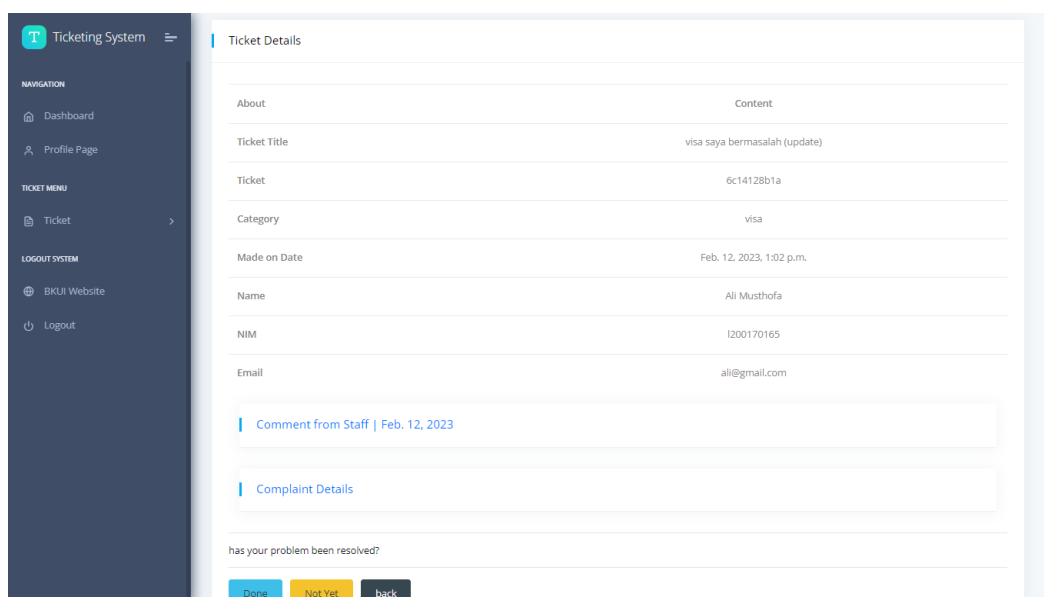
Halaman *All Ticket* mahasiswa ada tombol alert ketika tiket yang diinputkan mahasiswa sudah mendapatkan status finish. Maka mahasiswa dapat melihat detail tiket yang mereka inputkan serta dapat melihat komentar dari staf yang menangani tiket yang diinputkan, Gambar 19 merupakan visualisasi tampilan halaman *view all ticket* mahasiswa



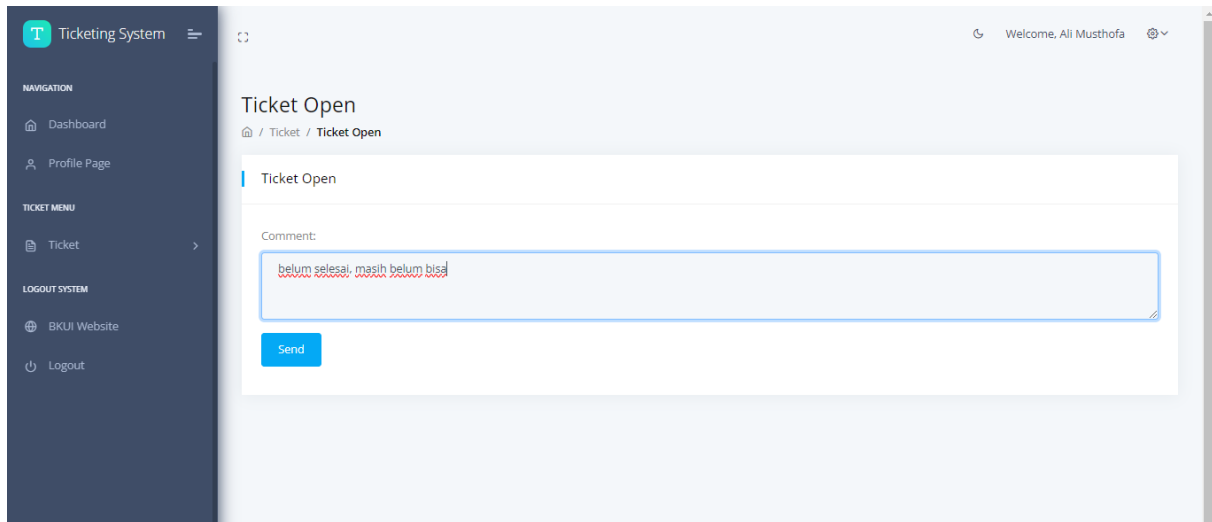
Gambar 20. Tampilan Halaman Semua Tiket Mahasiswa

3.1.9. Tampilan Halaman *Detail Ticket Confirmation*

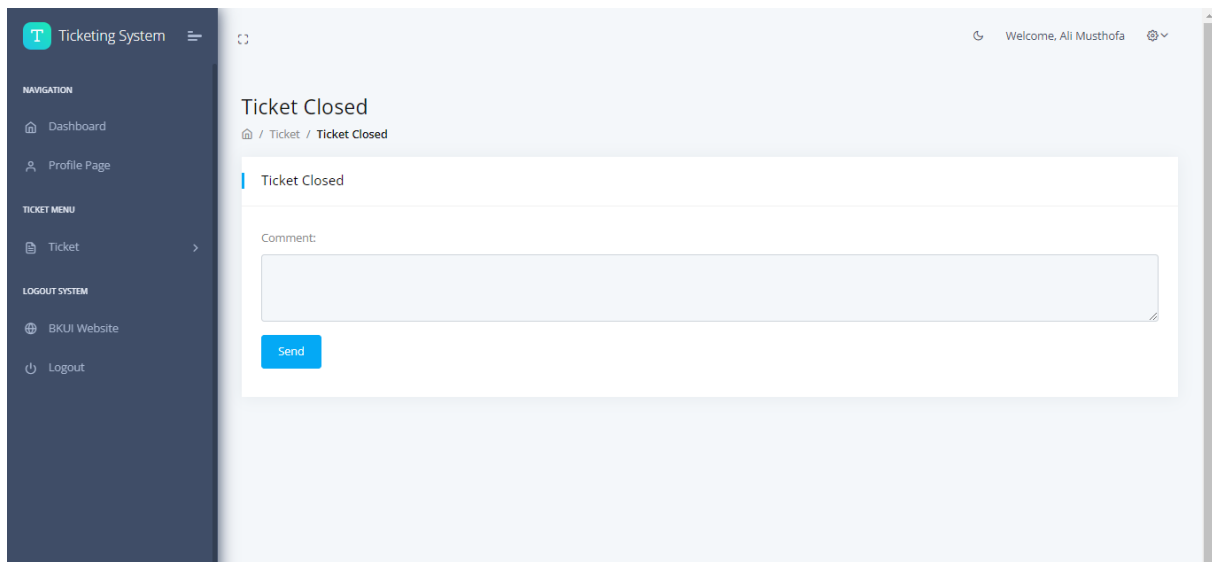
Halaman ini menampilkan detail tiket yang diinputkan serta halaman ini juga bisa dapat dikatakan konfirmasi kepada mahasiswa untuk melakukan konfirmasi apakah tiket sudah selesai, ketika mahasiswa merasa belum selesai, maka akan ada tombol *not yet* untuk membuka kembali tiket dengan status *open*. Tetapi ketika permasalahan sudah selesai maka mahasiswa dapat memilih tombol *done* dengan demikian mahasiswa tiket akan berubah status menjadi *closed*. Masing-masing tombol *done* dan *not yet* akan diteruskan ke halaman baru untuk membuat komentar, Gambar 21-23 merupakan visualisasi tampilan halaman *detail ticket confirmation*.



Gambar 21. Tampilan Halaman Konfirmasi Tiket



Gambar 22. Tampilan Halaman untuk membuka kembali tiket



Gambar 23. Tampilan Halaman untuk menutup Tiket

3.2. Pengujian dan Pembahasan

Pengujian sistem merupakan investigasi yang dilakukan untuk mendapatkan informasi tentang kualitas aplikasi. Melalui pengujian, suatu perangkat lunak dapat dilihat secara objektif dan independen serta risiko pada tahap implementasi dapat dipahami.

3.2.1. Pengujian *Black Box*

Tahap pengujian sistem merupakan tahap terakhir yang berfokus pada hasil akhir dan fungsi dari sistem. Pengujian *black box* adalah pengujian dengan sistem yang berinteraksi langsung dengan pengguna dan sistem mampu merespon dengan benar dan sesuai dengan yang diharapkan.

Tabel 1. *Black Box Testing*

Fungsi	Input	Output	Status
Halaman login	Masukkan email dan password yang benar	Menuju halaman dashboard atau home	Valid
Halaman profil	Jika ada form yang kosong, menginputkan data yang dibutuhkan	Menampilkan data yang diinputkan	valid
Halaman <i>add</i> tiket	Menginputkan data jenis aduan, aduan, kategori, serta file	Menuju halaman detail dan menampilkan data yang dimasukkan	valid
Halaman <i>update</i> tiket	Melakukan perubahan pada data untuk jenis aduan, aduan, kategori, serta file	Menuju halaman detail dan menampilkan data setelah di <i>update</i>	valid
Halaman <i>view</i>	Mengakses halaman view	Menampilkan tiket sesuai dengan permission	valid

3.2.2. the System Usability Scale (SUS)

Proses pengujian untuk penelitian *ticketing system* meliputi pengujian dengan *System Usability Scale* (SUS). Sistem ini setidaknya memiliki 4 aktor, yaitu Kaur (Kepala Urusan), staf, *front desk* dan mahasiswa, serta 12 pengguna berpartisipasi dalam pengujian sistem yang akan dikembangkan, dengan masing-masing pengguna dibagi menjadi 4 aktor.

Tabel 2. *System Usability Scale* (SUS) (Brooke, 1986)

No.	Penyataan
1	<i>I think that I would like to use this system frequently</i>
2	<i>I found the system unnecessarily complex</i>
3	<i>I thought the system was easy to use</i>
4	<i>I think that I would need the support of a technical person to be able to use this system</i>
5	<i>I found the various functions in this system were well integrated</i>
6	<i>I thought there was too much inconsistency in this system</i>

7	<i>I would imagine that most people would learn to use this system very quickly</i>
8	<i>I found the system very cumbersome to use</i>
9	<i>I felt very confident using the system</i>
10	<i>I needed to learn a lot of things before I could get going with this system</i>

Keterangan :

SS = Sangat Setuju (skor 5)

TS = Tidak Setuju (skor 2)

S = Setuju (skor 4)

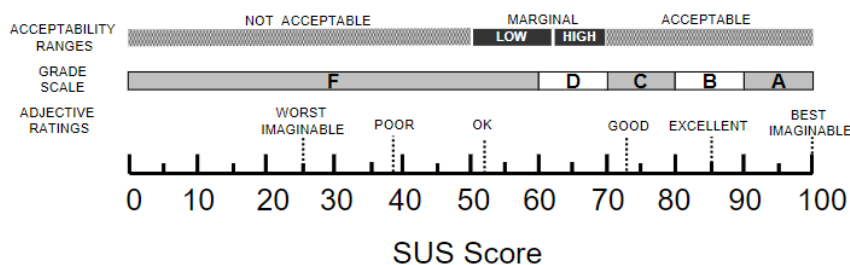
STS = Sangat Tidak Setuju (skor 1)

N = Netral (skor 3)

Hitung skor SUS dengan terlebih dahulu menambahkan skor untuk setiap item. Setiap item memiliki skor 0-4. Skor item 1, 3, 5, 7, dan 9 adalah posisi skala dikurangi 1. Skor item 2, 4, 6, 8, dan 10 adalah 5 dikurangi posisi skala. Kalikan jumlah poin dengan 2,5 untuk mendapatkan skor total SUS.

Untuk Skor SUS = (((S1-1)+(5-S2)+(S3-1)+(5-S4)+(S5-1)+(5-S6)+(S7-1)+ (5-S8)+(S9-1)+(5-S10)))*2.5)

Berdasarkan perhitungan dari skor SUS yang sudah didapatkan, pengkategorian sistem dinyatakan diterima atau tidak diterima dapat dilihat di Gambar 23.



Gambar 24. Perbandingan peringkat kata sifat, skor penerimaan, dan skala penilaian sekolah, dalam kaitannya dengan skor SUS rata-rata (Bangor et al., 2009)

Tabel 3. Hasil perhitungan SUS kepada responden

Responden	Skor										jumlah	Nilai (jumlah x 2.5)
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		
1	3	1	4	1	4	3	5	1	5	1	34	85
2	5	4	3	3	3	3	4	3	3	3	22	55
3	4	2	4	1	5	1	5	2	4	3	33	82,5
4	2	4	2	4	4	3	2	4	3	4	14	35
5	4	4	4	4	4	4	4	4	4	4	40	50

6	5	2	4	1	4	3	4	2	5	3	31	77,5
7	4	2	4	2	4	2	4	2	4	4	28	70
8	4	2	4	5	4	2	4	3	4	5	23	57,5
9	4	2	5	1	4	1	5	1	5	2	36	90
10	3	1	5	3	4	2	5	1	5	2	33	82,5
11	4	2	5	2	5	1	4	2	5	2	34	85
12	4	4	5	4	5	4	5	5	4	5	21	52,5
Average											69	

Hasil pengujian yang disajikan pada Tabel 3 menunjukkan bahwa hasil pengujian SUS adalah 69, berdasarkan Gambar 23 dapat disimpulkan bahwa sistem informasi yang dirancang dinilai baik dan dapat diterima.

4. PENUTUP

Setelah dilakukan perencanaan, implementasi dan pengujian sistem informasi *ticketing system*, maka ditarik beberapa kesimpulan. Pada saat pengujian sistem untuk mengukur kepuasan pengguna terhadap penggunaan *ticketing system* dengan kuesioner SUS diperoleh skor rata-rata 69 untuk 12 responden, sehingga dapat disimpulkan bahwa *ticketing system* dinilai baik dan dapat diterima. Pengujian sistem dengan *black box* berhasil sehingga semua langkah selesai dan semua fungsi sistem bekerja sesuai dengan yang diinginkan. Sistem informasi *ticketing system* untuk mahasiswa internasional dapat bertindak sebagai jembatan antara mahasiswa internasional dan staf jika mahasiswa mengalami masalah selama studi mereka. Semoga dengan sistem informasi *ticketing system* untuk pelayanan mahasiswa internasional ini dapat membantu staf dalam mengumpulkan, menyimpan dan mengolah informasi tentang mahasiswa internasional terkait permasalahan yang mereka hadapi. Sistem ini dibuat oleh penulis dengan kemampuan terbaiknya, namun masih perlu pengembangan lebih lanjut. Saran untuk pengembangan sistem seperti kebutuhan fungsi terjemahan atau tersedianya dua bahasa, Inggris dan Indonesia, aplikasi telepon seluler juga tersedia dalam sistem informasi untuk memfasilitasi pengaduan mahasiswa.

DAFTAR PUSTAKA

- Al-Hawari, F., & Barham, H. (2021). A machine learning based help desk system for IT service management. *Journal of King Saud University - Computer and Information Sciences*, 33(6), 702–718. <https://doi.org/10.1016/j.jksuci.2019.04.001>
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS. *Journal of Usability Studies*, 4(3), 114–123.

- Beisse, F. (2015). *A Guide to Computer User Support for Help Desk & Support Specialists*, (6th ed.). Cengage Learning.
- Brooke, J. (1986). SUS - A quick and dirty usability scale. *Usability Evaluation in Industry*, 189–194.
- Husen, M. F. (2020). *Pengembangan Aplikasi Sistem Pelaporan Helpdesk Sihepi Berbasis Website Pada Perum Percetakan Uang Republik Indonesia*. Universitas Muhammadiyah Surakarta.
- Mudhar, R. (2015). *Pembangunan sistem informasi helpdesk ticketing system menggunakan django framework(Studi kasus : SMK Saradan)*. Universitas Islam Negeri Syarif Hidayatullah Jakarta.
- Nidhra, S., & Dondeti, J. (2012). BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW. *International Journal of Embedded Systems and Applications (IJESA)* , Vol.2,(No.2), 29–50.
- Qoyyimah, Hidayah, N. A., & Fananie, Z. B. (2012). Rancang Bangun Helpdesk Ticketing System (STUDI KASUS: PT. PRIMUS INDOJAYA). *Studia Informatika: Jurnal Sistem Informasi* , 5(1), 1–7.
- Rachmawati, E., & Suhendra. (2018). Web-Based Ticketing System Helpdesk Application Using CodeIgniter Framework (Case Study : PT. Commonwealth Life). *International Journal of Computer Science and Mobile Computing*, , Vol.7(Issue.12), 29–41.
- Sommerville, I. (2011). *SOFTWARE ENGINEERING* (Ninth Edition). Addison-Wesley.
- Ward, J., & Peppard, J. (2002). *Strategic Planning for Information Systems* (Third Edition). John Wiley & Sons, LTD.
- Whitten, J. L., & Bentley, D. L. (2007). *Systems Analysis and Design Methods* (Seventh Edition). McGraw-Hill Education.