

BAB IV

PERANCANGAN

4.1 Analisis Sistem Terdahulu

Analisis sistem terdahulu bertujuan untuk menganalisis kondisi pada rancangan sistem yang dikembangkan. Pada tahap mengidentifikasi rancangan sistem, peneliti dapat menemukan apa yang dibutuhkan sistem berdasarkan pengamatan yang ada di lapangan. Analisis sistem terdahulu dapat memastikan perubahan pada sistem yang dilakukan untuk menutupi kekurangan dari sistem sebelumnya. MySAU merupakan sistem yang sebelumnya sudah dirancang untuk mempermudah pekerja dalam melakukan presensi jam masuk dan pulang yang sudah terintegrasi dengan *database*. Dalam tahap mengidentifikasi MySAU, peneliti mengusulkan untuk mengembangkan fitur baru, yaitu *Helpdesk* yang digunakan sebagai media koordinasi dalam mengajukan permintaan dari pekerja yang membutuhkan suatu sistem sebagai penunjang dalam memudahkan proses pekerjaan setiap pekerja. Pada saat ini di PT SAU, sistem *Helpdesk* yang digunakan masih dilakukan secara manual. Pekerja biasanya saat mengajukan *request* terkait keluhan atau permasalahan yang berkaitan dengan lingkup IT, masih menghubungi melalui telepon, *email*, *WhatsApp*, atau bicara secara langsung ke Staf IT. Saat pekerja mengajukan *request*, Staf IT dapat menerima *request* tersebut. Ketika data *request* telah diterima dan dikerjakan oleh Staf IT, pekerja tidak dapat memonitoring data *request* yang telah dikirim. Ketika data *request* selesai, pekerja akan dihubungi oleh Staf IT bahwa *request* yang telah diajukan tadi telah selesai. Data-data yang sebelumnya sudah dikerjakan oleh Staf IT, sangat sulit untuk ditemukan kembali. Oleh karena itu peneliti akan mengembangkan fitur *Helpdesk* ini di aplikasi MySAU untuk memudahkan staf IT dalam memonitoring data-data *request* tersebut.

4.2 Spesifikasi Kebutuhan Sistem Baru

Spesifikasi kebutuhan sistem baru membantu peneliti untuk menyelesaikan permasalahan. Selain dapat menyelesaikan permasalahan, kebutuhan sistem baru juga dapat membantu proses pembuatan aplikasi supaya tujuan penelitian ini dapat tercapai.

1) Spesifikasi Perangkat Keras (*Hardware*)

Perangkat keras merupakan perangkat yang tampak secara fisik yang terdiri dari perangkat *input*, *output*, *data processing*, *communication*, dan *data storage*. Pada penelitian ini, spesifikasi perangkat keras sangat diperlukan untuk mendukung pengoperasian perangkat lunak dan perangkat keras, sebagai berikut.

Tabel 4. 1 Spesifikasi *Hardware*

No.	Perangkat Keras	Keterangan
1.	Processor	Intel i7-9750H
2.	RAM	8 GB RAM
3.	SSD	512 GB SSD
4.	<i>Graphic Card</i>	NVIDIA GeForce GTX 1650 TI

2) Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak dapat mengendalikan perangkat keras, dan perangkat lunak aplikasi merupakan pondasi dari sistem komputer agar dapat memberikan layanan yang dibutuhkan. Perangkat lunak dibutuhkan juga dalam pengembangan aplikasi, sebagai berikut.

Tabel 4. 2 Spesifikasi *Software*

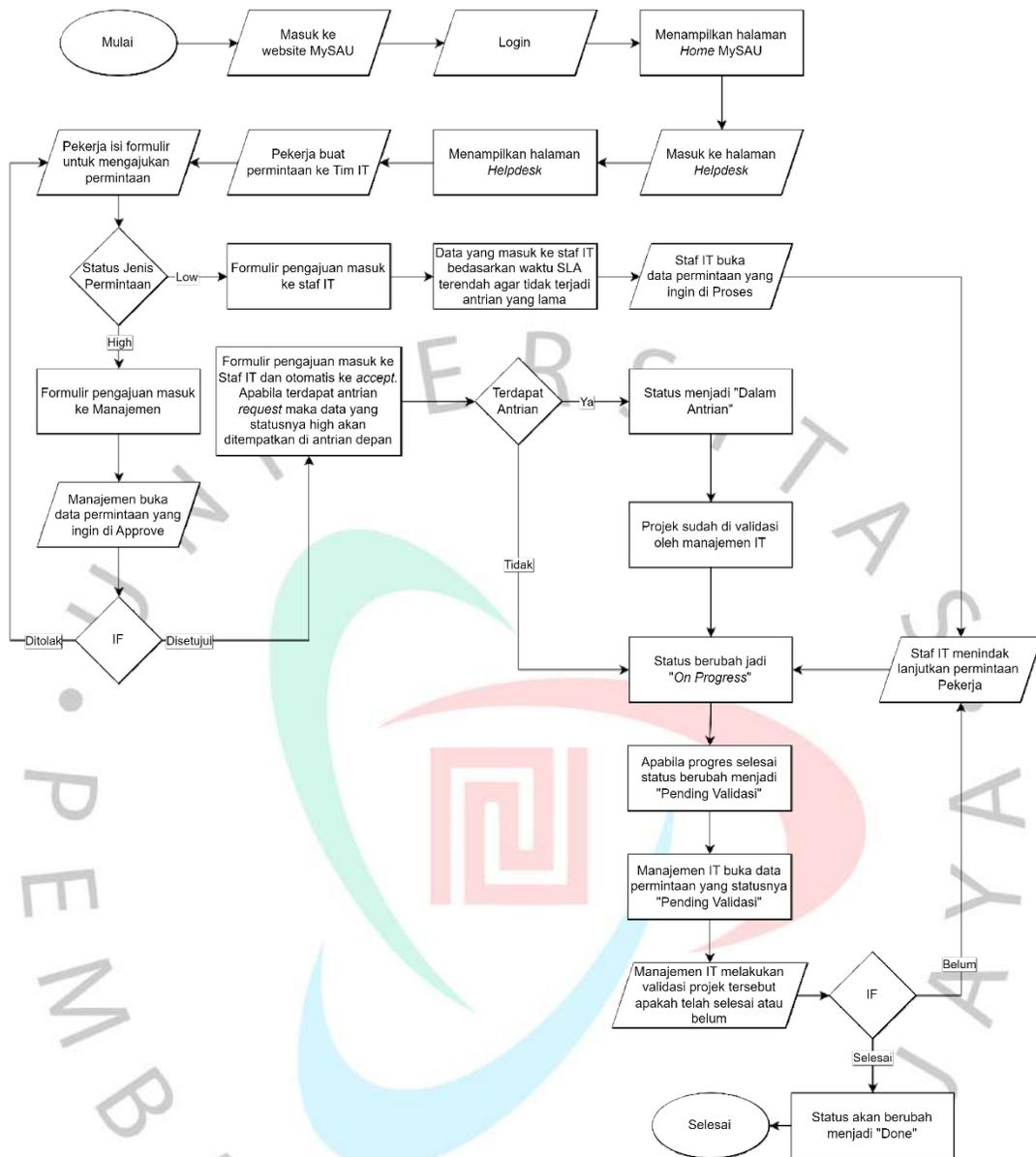
No.	Perangkat Lunak
1.	Sistem Operasi Windows 11
2.	Google Chrome
3.	Visual Studio Code
4.	Laragon

3) Proses

Proses yang akan dilakukan dalam pengembangan sistem aplikasi MySAU pada fitur *Helpdesk* yaitu pekerja yang membutuhkan bantuan yang berkaitan dengan IT ke tim IT harus mengajukan permintaan melalui aplikasi MySAU tersebut. Pekerja harus masuk ke webiste MySAU dan login melalui *Personal Number* (PN) pekerja. Setelah login pekerja masuk ke menu *Helpdesk*, dan mengajukan permintaan yang diinginkan dengan mengisi *form request* untuk mengajukan permintaan. Saat pekerja mengisi *form request* akan terdapat kondisi terdapat jenis permintaan tersebut apakah prioritas atau tidak, permintaan yang di prioritaskan akan terdapat *approver* dan yang tidak diprioritaskan tidak terdapat *approver*. *Form request* yang telah diisi oleh pekerja dan

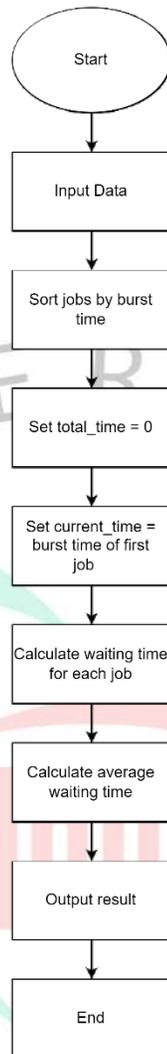
telah dikirim, maka data akan masuk ke manajemen. Sedangkan permintaan yang tidak membutuhkan *approver*, maka data yang dikirim akan masuk ke staf IT.

Pada data permintaan yang membutuhkan *approval* dari manajemen jika permintaan tersebut ditolak, maka pekerja harus mengirim ulang permintaan pengajuan dan apabila disetujui maka data tersebut otomatis di *accept* dengan status '*On Progress*' lalu dikerjakan oleh staf IT. Pada data yang masuk ke tim staf IT, maka staf IT akan buka data permintaan tersebut yang akan diproses. Setelah permintaan telah diproses maka status akan berubah menjadi '*Pending Validasi*'. Data permintaan yang telah dikerjakan oleh staf IT, maka permintaan tersebut membutuhkan validasi dari manajemen IT untuk mengecek apakah proyek tersebut sudah selesai atau belum. Ketika proyek yang telah diajukan oleh pekerja sudah selesai, maka permintaan tersebut akan di *approve* manajemen IT dan telah dinyatakan telah selesai. Terdapat hasil *report* dari fitur *Helpdesk* tersebut berupa data *request* yang telah melakukan pengajuan oleh pekerja. Data *report* tersebut di tampilkan berupa data statistik yang dimana isi data tersebut adalah jenis permintaan yang paling banyak dibuat. Peneliti juga membuat diagram yang menggambarkan proses aplikasi MySAU pada fitur *Helpdesk*.



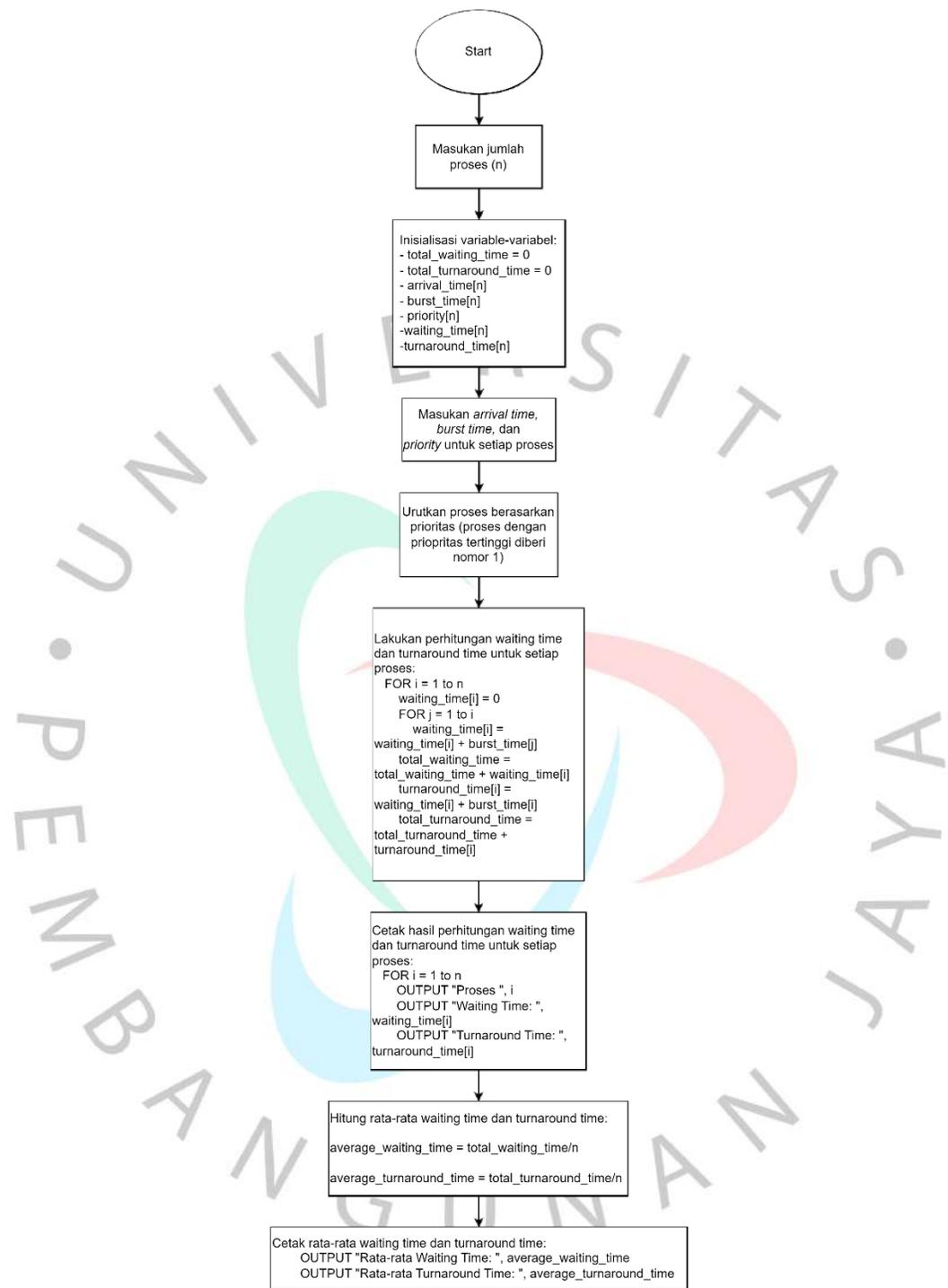
Gambar 4. 1 Alur Proses

Selain alur proses aplikasi, peneliti membuat alur proses algoritma *shortest job first* dan *priority scheduling* sebagai berikut.



Gambar 4. 2 Alur Algoritma Shortest Job First

Pada gambar () merupakan alur algoritma *shortest job first*. Pertama pada tahap *input* data berisi pengumpulan informasi pekerjaan yang akan diproses seperti *burst time*. Selanjutnya merupakan langkah untuk mengurutkan pekerjaan berdasarkan waktu pemrosesan yang paling pendek. Pada tahap Set total_time = 0 digunakan untuk menginisialisasi variabel total_time sebagai waktu total yang telah berlalu. Set current_time = *burst time* juga digunakan untuk menginisialisasi variabel total_time dan current_time dengan nilai awal. Pada tahap selanjutnya merupakan langkah untuk menghitung waktu tunggu dari setiap pekerjaan. Lalu setelah waktu tunggu, bisa dilakukan perhitungan untuk mendapatkan rata-rata waktu tunggu. Setelah itu menampilkan hasil akhir dari proses algoritma *shortest job first*.



Gambar 4. 3 Alur Algoritma Priority Scheduling

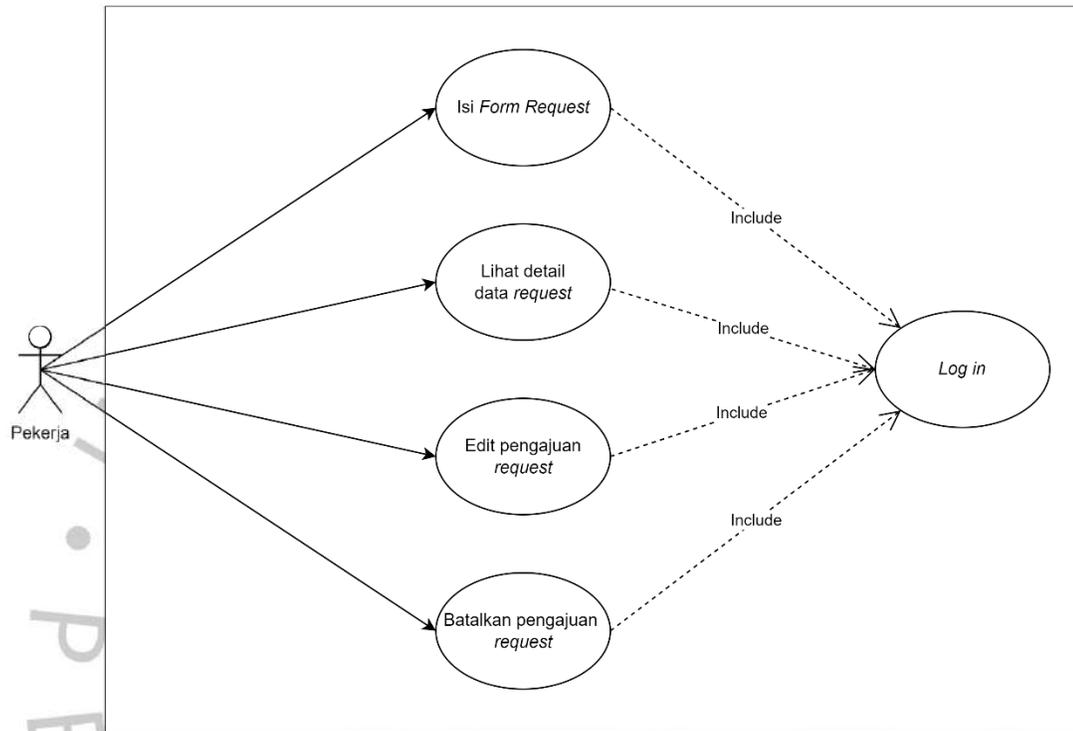
Pada gambar () merupakan alur algoritma *priority scheduling*. Langkah pertama meminta *user* untuk memasukkan jumlah proses yang akan dijadwalkan. Selanjutnya variabel-variabel yang akan digunakan dalam algoritma diinisialisasi. Dalam hal ini

termasuk variabel untuk menghitung total *waiting time* dan total *turnaround time*, serta variabel untuk menyimpan *arrival time*, *burst time*, *priority*, *waiting time*, dan *turnaround time* untuk setiap proses. *User* akan diminta untuk memasukkan nilai *arrival time*, *burst time*, dan *priority* untuk setiap proses yang akan dijadwalkan. Proses diurutkan berdasarkan nilai prioritas, proses dengan prioritas yang lebih tinggi akan diberi nomor 1. Tahap selanjutnya melibatkan perhitungan *waiting time* dan *turnaround time* untuk setiap proses. *Waiting time* adalah total waktu yang dihabiskan oleh proses sebelum dapat memulai dieksekusi, sedangkan *turnaround time* adalah waktu yang dibutuhkan oleh proses mulai dari saat kedatangan hingga selesai dieksekusi. Setelah melibatkan perhitungan *waiting time* dan *turnaround time*, hasil perhitungan dicetak untuk keperluan informasi dan analisis. Total *waiting time* dan *turnaround time* dihitung untuk semua proses, kemudian rata-rata *waiting time* dan *turnaround time* dihitung dengan membagi total dengan jumlah proses. Rata-rata *waiting time* dan *turnaround time* dicetak untuk memberikan gambaran tentang jinerja keseluruhan algoritma *priority scheduling* yang digunakan.

4.3 Perancangan Sistem

Proses perancangan sistem merupakan memahami sistem kemudian membuat sistem informasi berbasis komputer, hasil akhirnya yaitu sistem yang terkomputerisasi. Pada penelitian ini, peneliti menjelaskan perancangan sistem pada aplikasi yang dikembangkan dengan model *Unified Modeling Language* (UML) dengan 4 metode yaitu *use case diagram*, *activity diagram*, *sequence diagram*, dan *entity relationship diagram*. Berikut adalah rancangan sistem yang dibuat oleh peneliti di setiap UML.

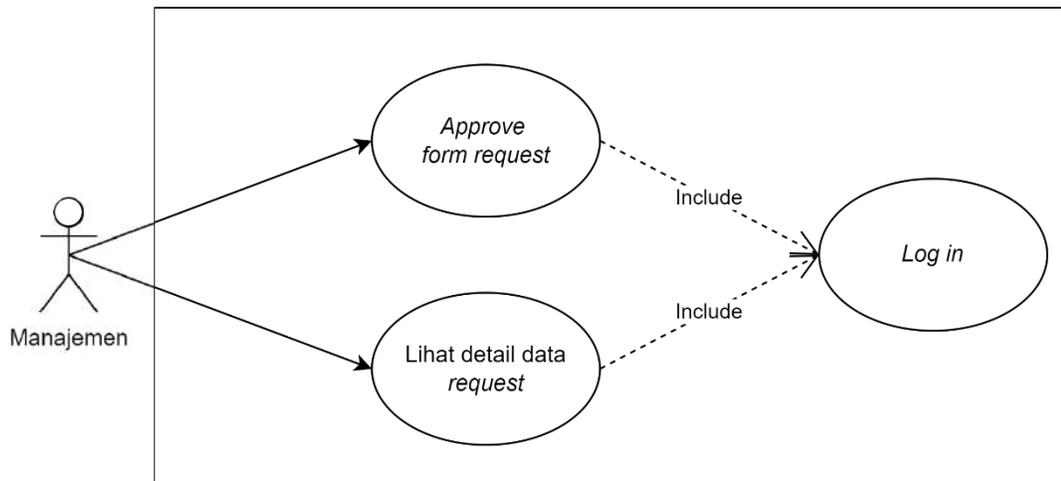
4.3.1 Use Case Diagram



Gambar 4. 4 Use Case Diagram Pekerja

Tabel 4. 3 Skenario Pekerja

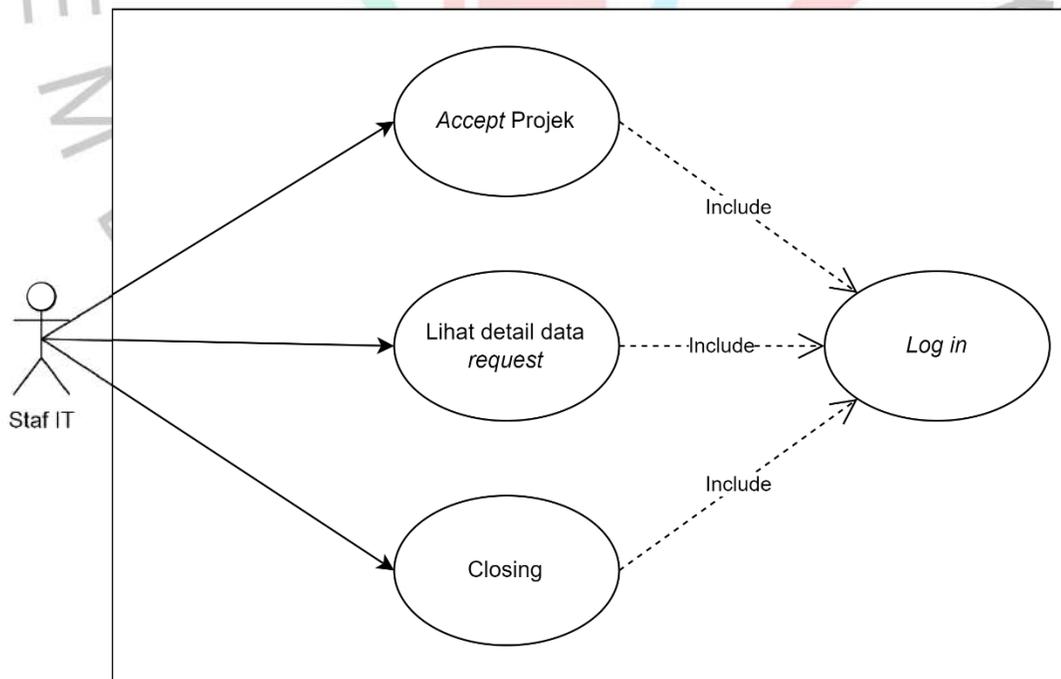
Nama Use Case	<i>Login</i>
Aktor	Pekerja
Deskripsi	Pekerja <i>login</i> ke aplikasi MySAU
Tindakan	Pekerja dapat membuat <i>form request</i> , update data <i>request</i> , <i>cancel</i> data <i>request</i> , dan dapat view detail data <i>request</i> .



Gambar 4. 5 Use Case Diagram Managemen

Tabel 4. 4 Skenario Managemen

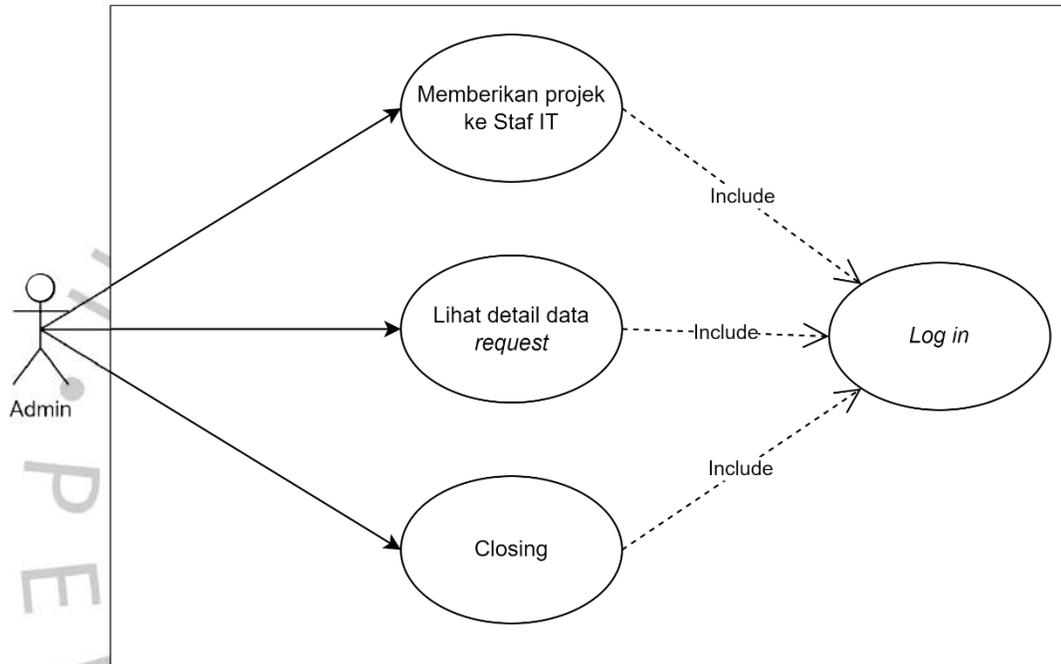
Nama Use Case	<i>Login</i>
Aktor	Managemen
Deskripsi	Managemen <i>login</i> ke aplikasi MySAU
Tindakan	Managemen dapat melakukan <i>approve form request</i> dan <i>view detail data request</i> .



Gambar 4. 6 Use Case Diagram Staf IT

Tabel 4. 5 Skenario Staf IT

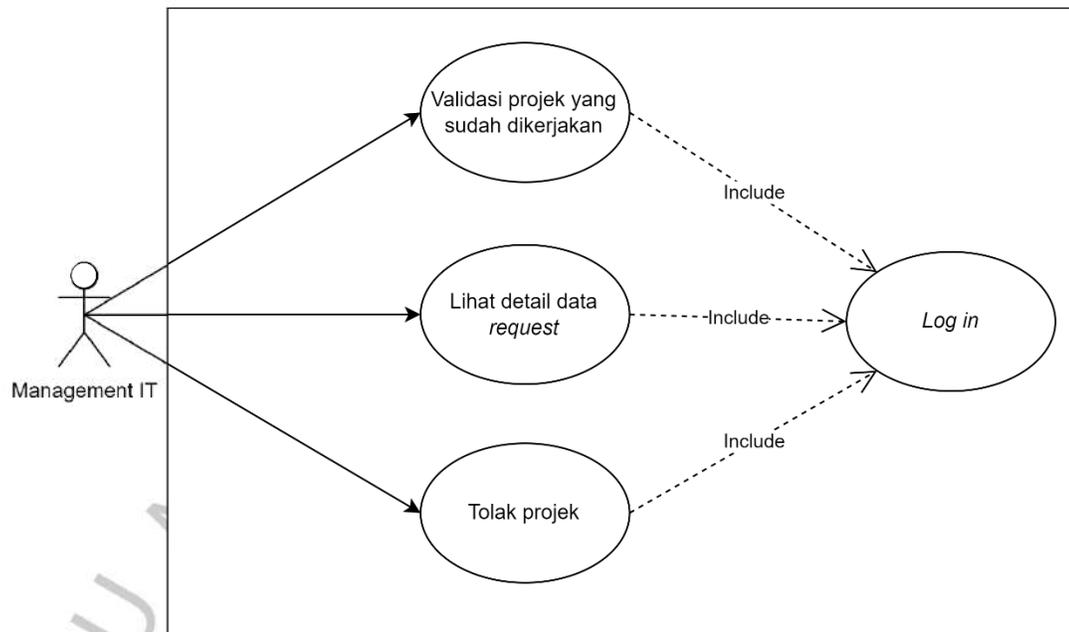
Nama Use Case	<i>Login</i>
Aktor	Staf IT
Deskripsi	Staf IT <i>login</i> ke aplikasi MySAU
Tindakan	Staf IT dapat melakukan <i>accept</i> projek, <i>view</i> detail data <i>request</i> , dan <i>closing</i> data <i>request</i> .



Gambar 4. 7 Use Case Diagram Admin

Tabel 4. 6 Skenario Admin

Nama Use Case	<i>Login</i>
Aktor	Admin
Deskripsi	Admin <i>login</i> ke aplikasi MySAU
Tindakan	Admin dapat melakukan <i>closing</i> dan <i>view</i> detail data <i>request</i> .

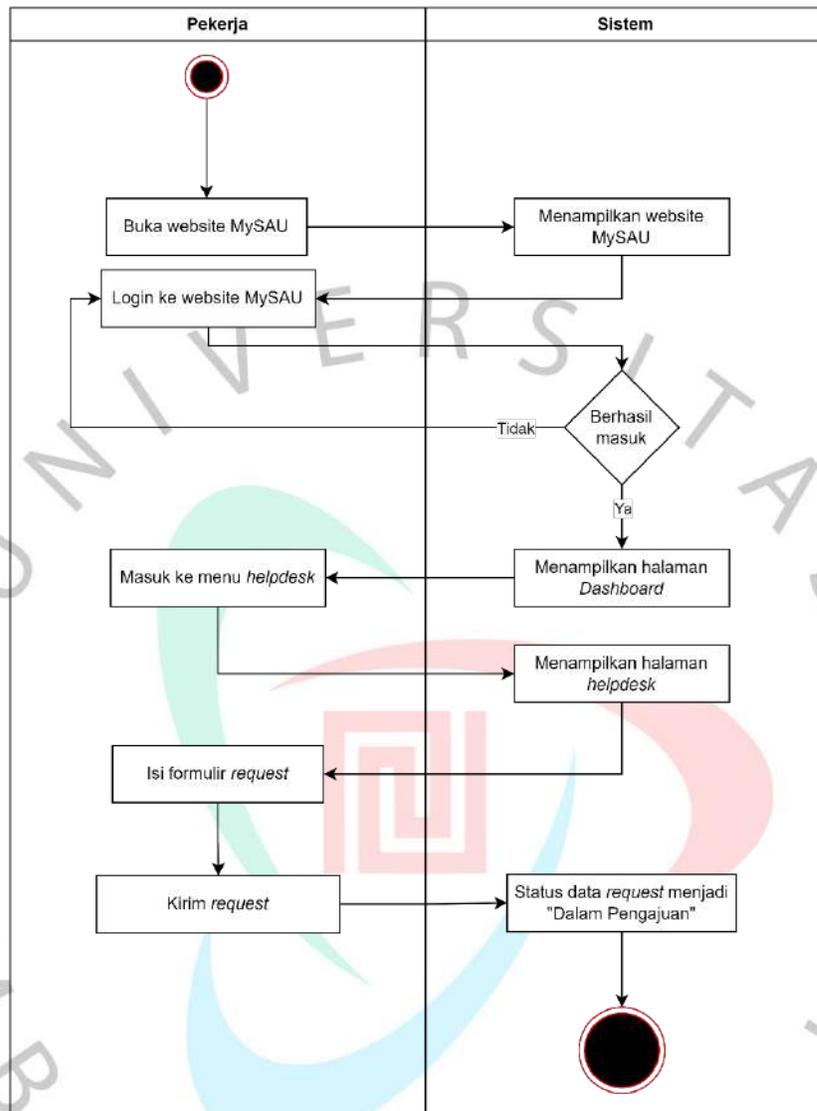


Gambar 4. 8 Use Case Diagram Manajemen IT

Tabel 4. 7 Skenario Manajemen IT

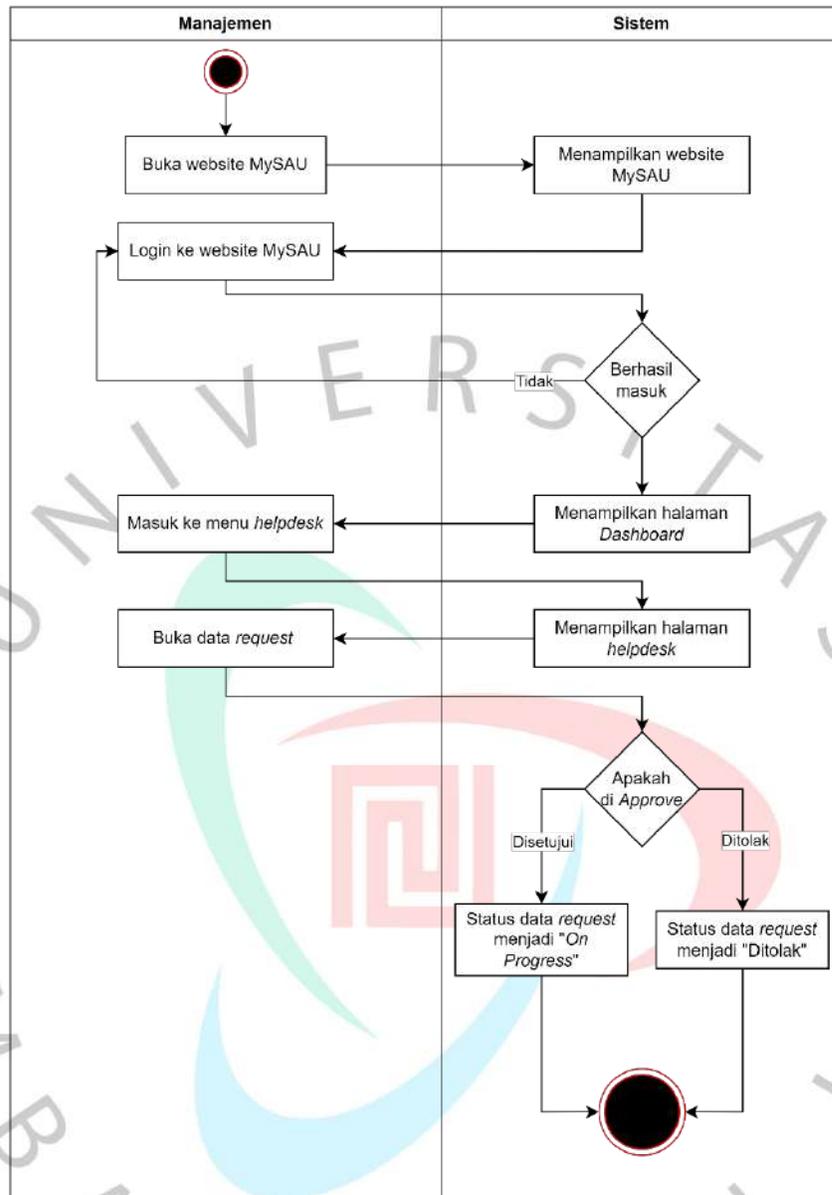
Nama Use Case	<i>Login</i>
Aktor	Manajemen IT
Deskripsi	Manajemen IT <i>login</i> ke aplikasi MySAU
Tindakan	Manajemen IT dapat melakukan validasi dan <i>view</i> detail data <i>request</i> .

4.3.2 Activity Diagram



Gambar 4. 9 Activity Diagram Pekerja

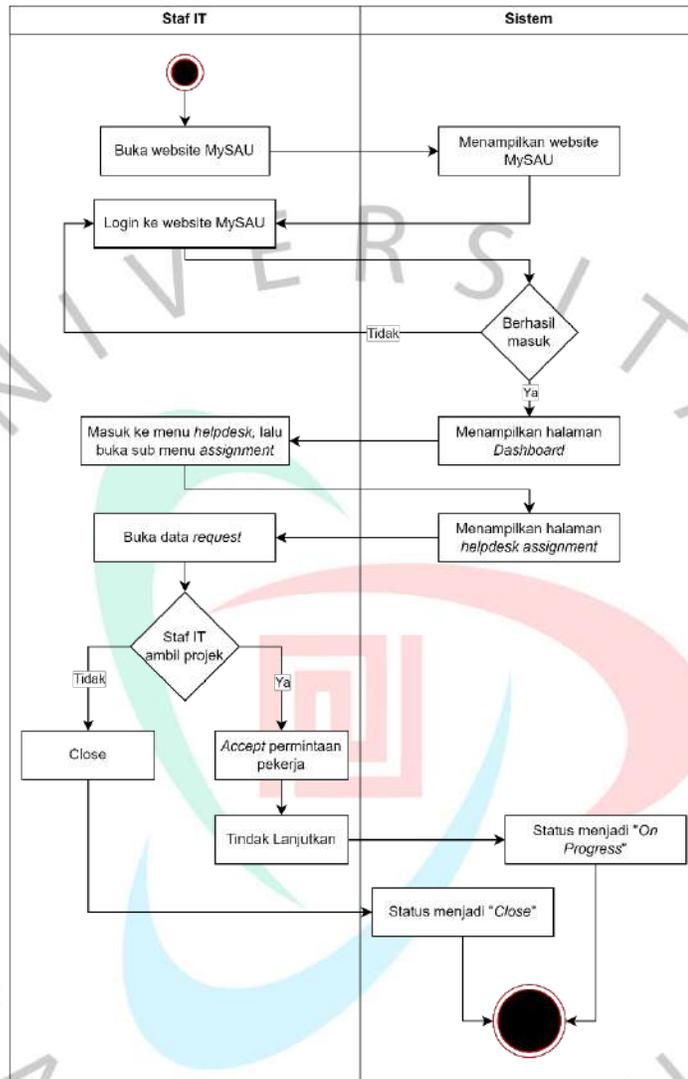
Pada gambar 4.7 merupakan *activity* diagram dari *role* pekerja. Pekerja harus melakukan login terlebih dahulu di website MySAU. Pekerja yang berhasil masuk ke website MySAU maka akan menampilkan halaman dashboard, apabila pekerja tidak dapat berhasil login maka harus melakukan login ulang atau masuk ke website MySAU. Setelah pekerja berhasil login dan masuk ke website MySAU, pekerja masuk ke menu *Helpdesk* lalu buat *request* kepada Staf IT dengan mengisi formulir. Pada saat pekerja telah mengisi formulir *request*, formulir tersebut yang telah dikirim statusnya akan menjadi “Dalam Pengajuan”.



Gambar 4. 10 Activity Diagram Managemen

Pada gambar 4.8 merupakan *activity* diagram dari *role* Managemen. Managemen melakukan login terlebih dahulu pada website MySAU. Managemen yang berhasil masuk ke website MySAU maka akan menampilkan halaman dashboard, apabila managemen tidak dapat berhasil login maka harus melakukan login ulang atau masuk ke website MySAU. Managemen yang telah berhasil login dan masuk ke website MySAU, maka dapat masuk ke menu *Helpdesk* lalu buka data *request*. Setelah data *request* sudah dibuka, managemen dapat menyetujui atau menolak data *request* yang telah diajukan oleh pekerja. Apabila data *request* disetujui, maka data statusnya “On

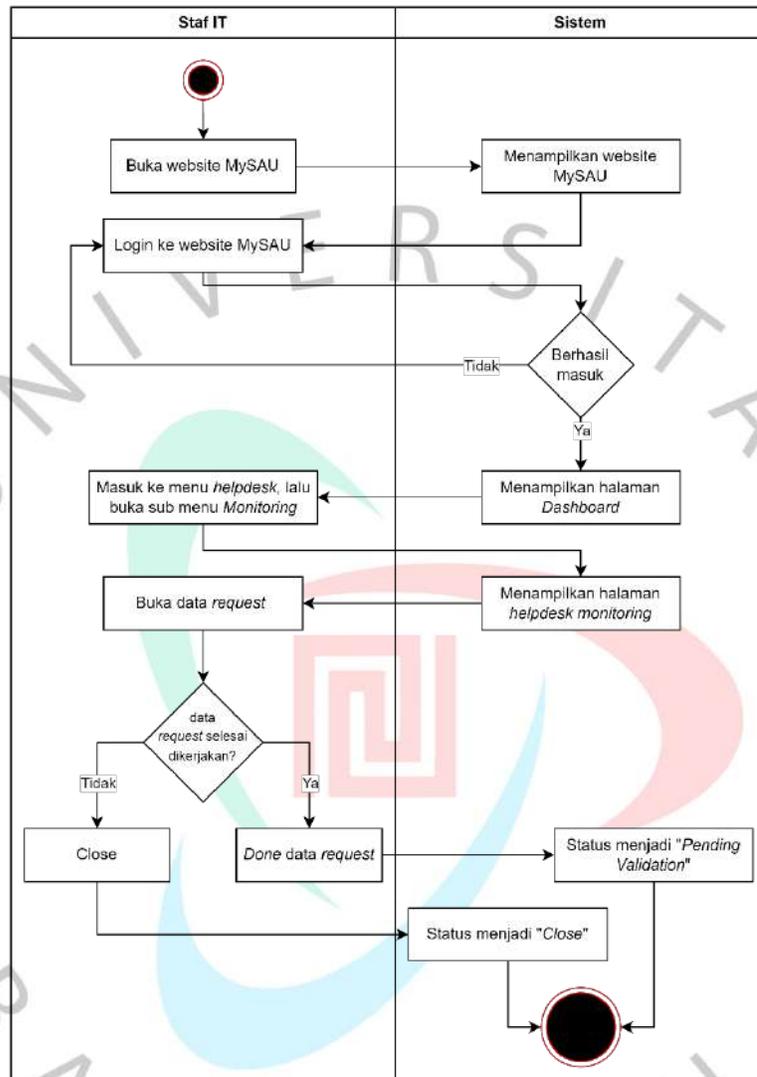
Progress”. Jika data *request* ditolak oleh manajemen, maka data tersebut statusnya “Ditolak” dan pekerja harus membuat data *request* ulang.



Gambar 4. 11 Activity Diagram Staf IT menu Assigment

Pada gambar 4.9 merupakan *activity* diagram dari *role* Staf IT. Staf IT login terlebih dahulu pada website MySAU. Staf IT yang berhasil masuk ke website MySAU maka akan menampilkan halaman dashboard, apabila staf IT tidak dapat berhasil login maka harus melakukan login ulang atau masuk ke website MySAU. Staf IT yang telah berhasil login dan masuk ke website MySAU, maka dapat masuk ke menu *Helpdesk* lalu buka sub menu *Assignment*. Staf IT buka data *request* yang sudah masuk, lalu data *request* tersebut dapat di ambil oleh Staf IT. Apabila data *request* tersebut di *accept* oleh staf IT, maka data *request* tersebut akan dikerjakan oleh staf IT dan statusnya

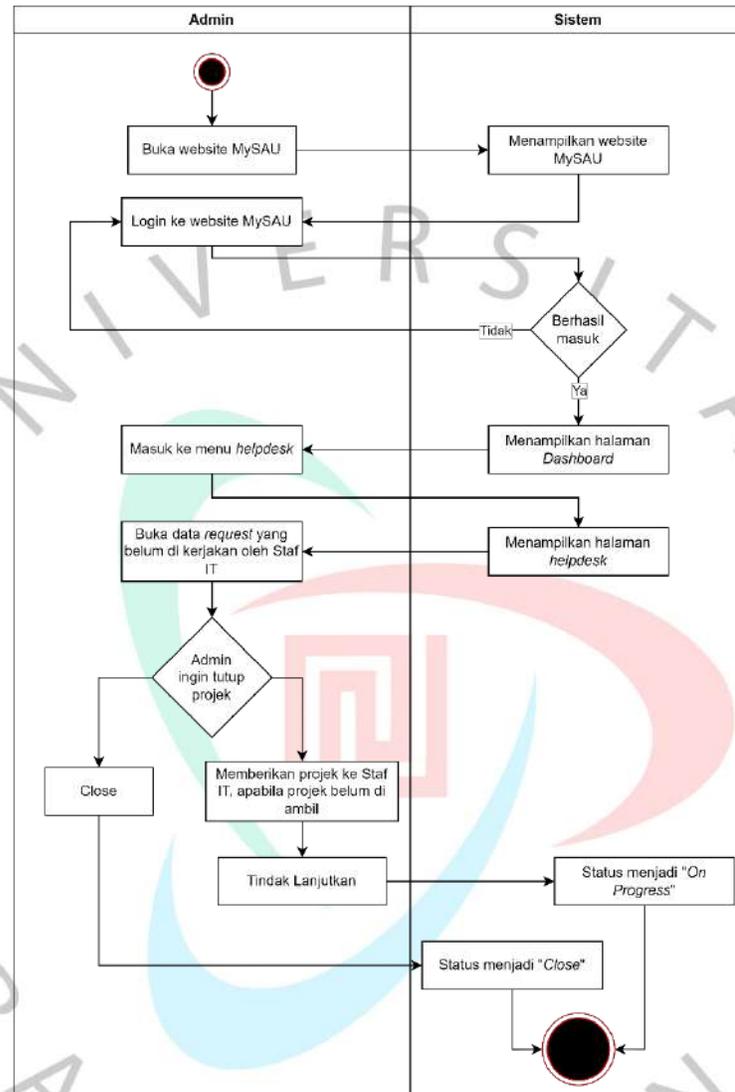
menjadi “*On Progress*”. Jika data *request* tersebut di *close* oleh Staf IT, maka data tersebut tidak akan dikerjakan oleh staf IT dan statusnya berubah menjadi “*Close*”.



Gambar 4. 12 Diagram Activity Staf IT menu Monitoring

Pada gambar 4.10 merupakan *activity* diagram dari *role* Staf IT. Staf IT login terlebih dahulu pada website MySAU. Staf IT yang berhasil masuk ke website MySAU maka akan menampilkan halaman dashboard, apabila staf IT tidak dapat berhasil login maka harus melakukan login ulang atau masuk ke website MySAU. Staf IT yang telah berhasil login dan masuk ke website MySAU, maka dapat masuk ke menu *Helpdesk* lalu buka sub menu *Monitoring*. Buka data *request* yang telah dikerjakan oleh staf IT, maka staf IT dapat *done* data *request* tersebut. Data *request* yang telah di *done* oleh staf

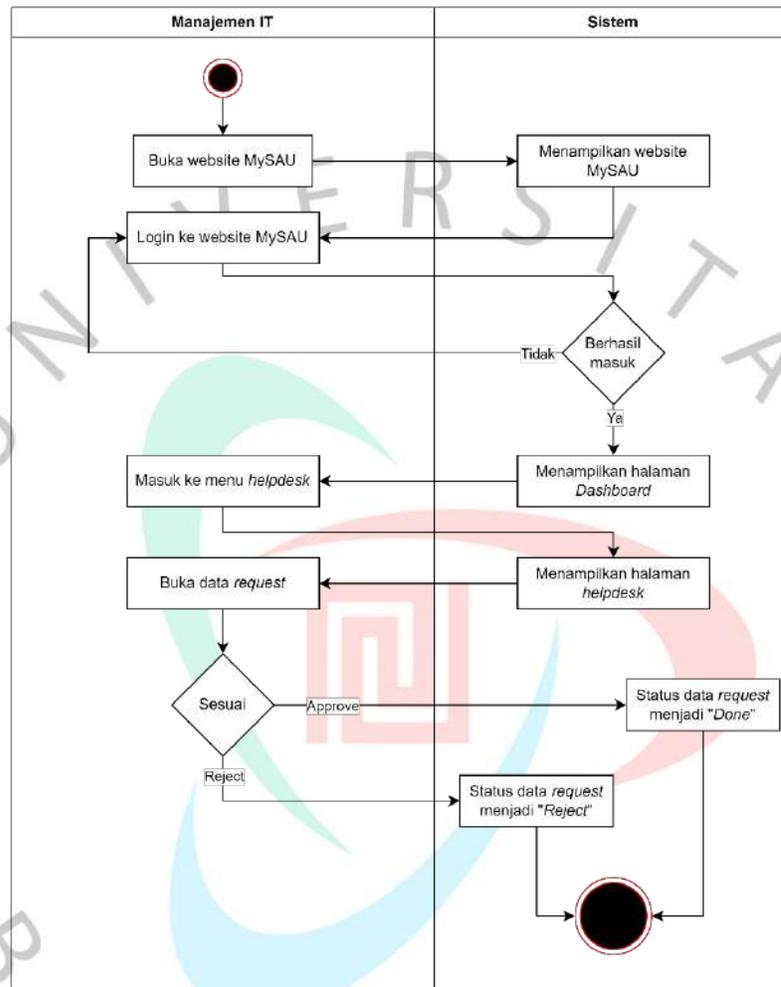
IT, maka statusnya akan menjadi “*Pending Validation*”. Apabila data tersebut di *close* oleh staf IT, maka statusnya akan menjadi “*Close*”.



Gambar 4. 13 Activity Diagram Admin

Pada gambar 4.11 merupakan *activity* diagram dari *role* Admin. Admin login terlebih dahulu pada website MySAU. Admin yang berhasil masuk ke website MySAU maka akan menampilkan halaman dashboard, apabila admin tidak dapat berhasil login maka harus melakukan login ulang atau masuk ke website MySAU. Admin yang telah berhasil login dan masuk ke website MySAU, maka dapat masuk ke menu *Helpdesk* lalu buka data *request* yang belum dikerjakan oleh staf IT. Admin dapat memberikan data *request* ke staf IT apabila data *request* tersebut belum di ambil oleh staf IT. Admin

juga dapat menutup data *request* pekerja. Data *request* yang telah diberikan ke staf IT, maka statusnya akan menjadi “*On Progress*”. Apabila data *request* tersebut di *close* oleh admin, maka statusnya menjadi “*Close*”.

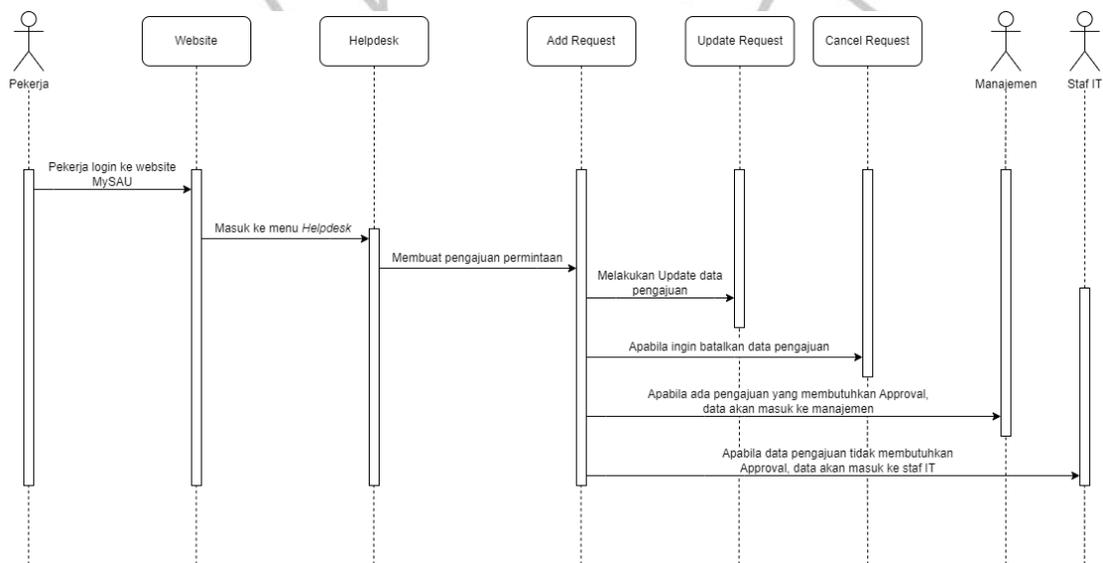


Gambar 4. 14 Activity Diagram Managemen IT

Pada gambar 4.12 merupakan *activity* diagram dari *role* Managemen IT. Managemen IT login terlebih dahulu pada website MySAU. Managemen IT yang berhasil masuk ke website MySAU maka akan menampilkan halaman dashboard, apabila managemen IT tidak dapat berhasil login maka harus melakukan login ulang atau masuk ke website MySAU. Managemen IT yang telah berhasil login dan masuk ke website MySAU, maka dapat masuk ke menu *Helpdesk* lalu buka data *request* yang telah dikerjakan oleh staf IT. Managemen IT melakukan validasi pada data *request* yang telah dikerjakan oleh staf IT, apakah data *request* tersebut sudah selesai dikerjakan dan

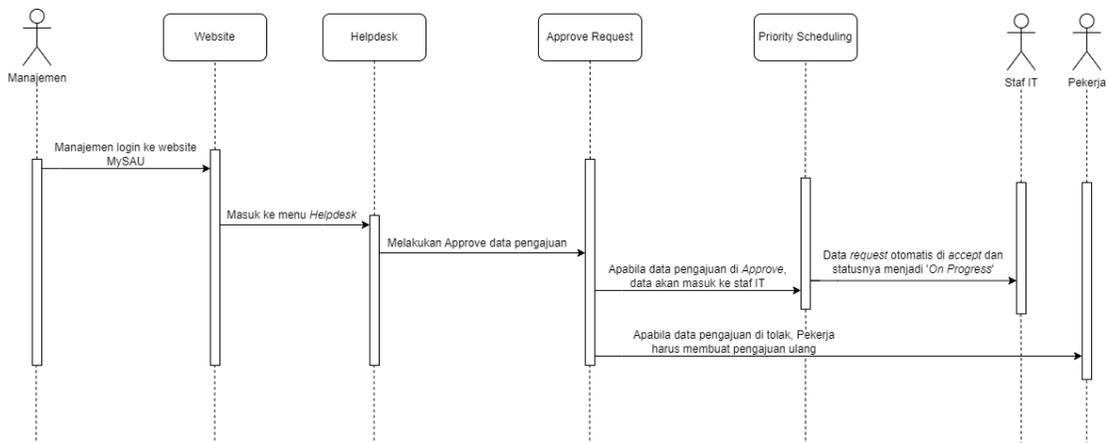
sesuai dengan permintaan pekerja atau data *request* tersebut belum selesai dikerjakan atau tidak sesuai dengan permintaan pekerja. Ketika manajemen IT validasi data *request* dan data tersebut sudah selesai atau sudah sesuai dengan permintaan pekerja, maka data *request* tersebut akan di *approve* oleh manajemen IT dan statusnya berubah menjadi "Done". Apabila data *request* tersebut belum selesai dikerjakan atau belum sesuai dengan permintaan pekerja, maka dan statusnya berubah menjadi "Reject".

4.3.3 Sequence Diagram



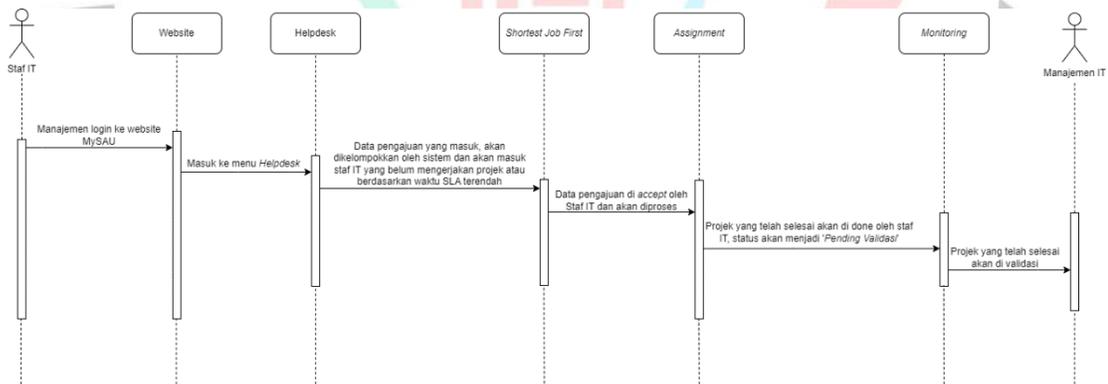
Gambar 4. 15 Sequence Diagram Pekerja

Pada gambar 4.13 merupakan *sequence* diagram dari *role* pekerja. Pertama pekerja melakukan login di website MySAU. Selanjutnya buka menu *Helpdesk*, lalu buat pengajuan permintaan. Setelah pekerja membuat data *request*, pekerja dapat melakukan *update* pada data *request* tersebut atau bisa juga *cancel* data *request* tersebut. Pekerja yang telah membuat data *request*, data tersebut akan dikirim ke manajemen atau staf IT. Data *request* yang dikirim ke manajemen merupakan data yang jenis permintaannya membutuhkan *approval*. Sedangkan data *request* yang dikirim ke staf IT merupakan data yang jenis permintaannya tidak membutuhkan *approval*.



Gambar 4. 16 *Sequence Diagram Managemen*

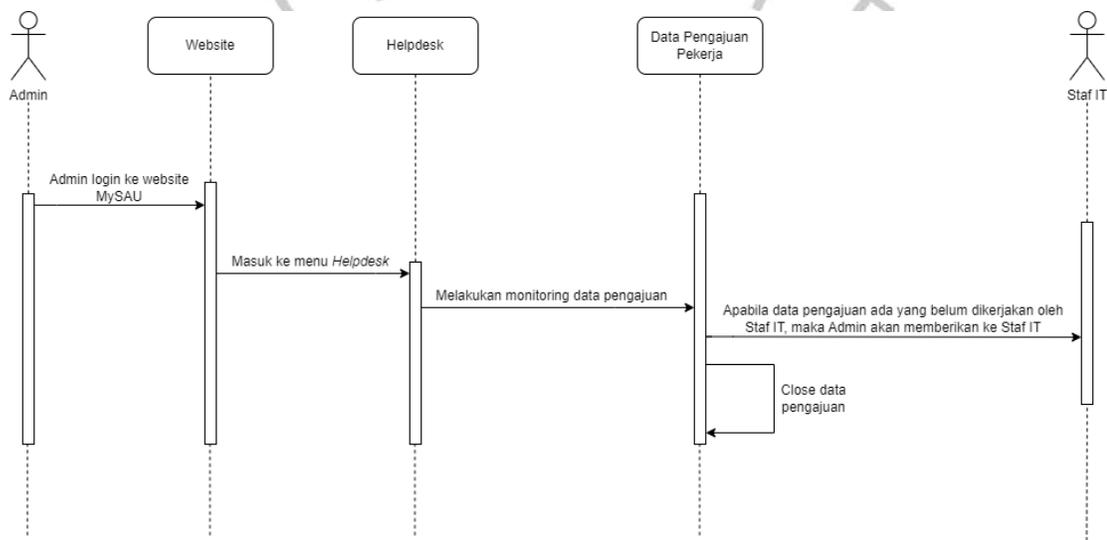
Pada gambar 4.14 merupakan *sequence diagram* dari *role* manajemen. Pertama pekerja melakukan login di website MySAU. Selanjutnya buka menu *Helpdesk*, lalu buka data *request* pekerja. Manajemen melakukan *approve* data *request* yang telah dikirim oleh pekerja. Apabila data *request* tersebut di *approve* oleh manajemen, maka data tersebut akan masuk ke staf IT. jika data *request* tersebut ditolak, maka pekerja harus mengajukan ulang data *request*.



Gambar 4. 17 *Sequence Diagram Staf IT*

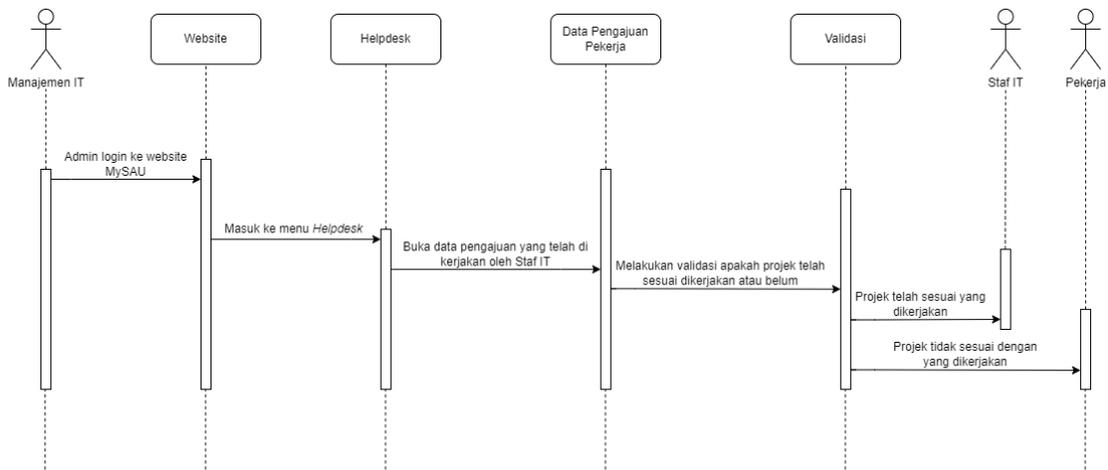
Pada gambar 4.15 merupakan *sequence diagram* dari *role* staf IT. Pertama pekerja melakukan login di website MySAU. Selanjutnya buka menu *Helpdesk*, data *request* yang telah dikirim oleh pekerja dan masuk ke staf IT, data tersebut akan dikelompokkan oleh sistem. Data tersebut akan masuk ke staf IT berdasarkan waktu *Service Level Agreement (SLA)*. Apabila ada staf IT yang belum mendapatkan atau mengerjakan data *request*, maka data tersebut akan masuk ke staf IT yang belum

mendapatkan atau mengerjakan data *request*. Atau data *request* tersebut dapat masuk ke staf IT yang waktu pengerjaannya paling sedikit atau mau selesai. Pada fitur *Helpdesk* tersebut menggunakan sistem pendukung keputusan (SPK) dengan algoritma *Shortest Job First* yang dimana data *request* dengan waktu SLA paling rendah akan dikerjakan duluan. Data *request* yang telah di *accept* oleh staf IT, maka data tersebut akan diproses oleh staf IT. Ketika data sudah selesai dikerjakan, maka data tersebut akan masuk ke manajemen IT untuk dilakukan validasi.



Gambar 4. 18 *Sequence* diagram Admin

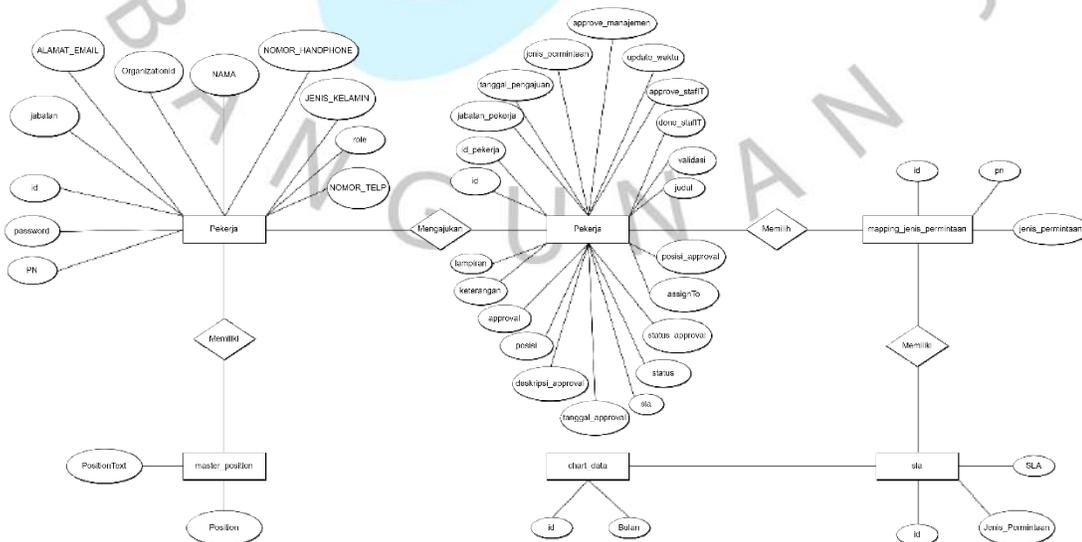
Pada gambar 4.16 merupakan *sequence* diagram dari *role* admin. Pertama pekerja melakukan login di website MySAU. Selanjutnya buka menu *Helpdesk*, lalu melakukan monitoring data *request*. Admin dapat memberikan data *request* ke staf IT apabila data tersebut belum dikerjakan oleh staf IT. Admin juga dapat melakukan *close* data *request*.



Gambar 4. 19 Sequence Diagram Manajemen IT

Pada gambar 4.16 merupakan *sequence* diagram dari *role* manajemen IT. Pertama pekerja melakukan login di website MySAU. Selanjutnya buka menu *Helpdesk*, lalu buka data *request* yang telah dikerjakan oleh staf IT. Setelah dibuka data *request* tersebut, manajemen IT dapat melakukan pengecekan pada data *request* yang telah dikerjakan oleh staf IT. Apabila data *request* tersebut sudah selesai dikerjakan dan sesuai dengan data *request* pekerja, maka data *request* tersebut dinyatakan telah selesai. Tetapi jika data *request* tersebut belum dikerjakan oleh staf IT atau tidak sesuai dengan data *request* pekerja, maka data tersebut akan dikerjakan kembali oleh staf IT.

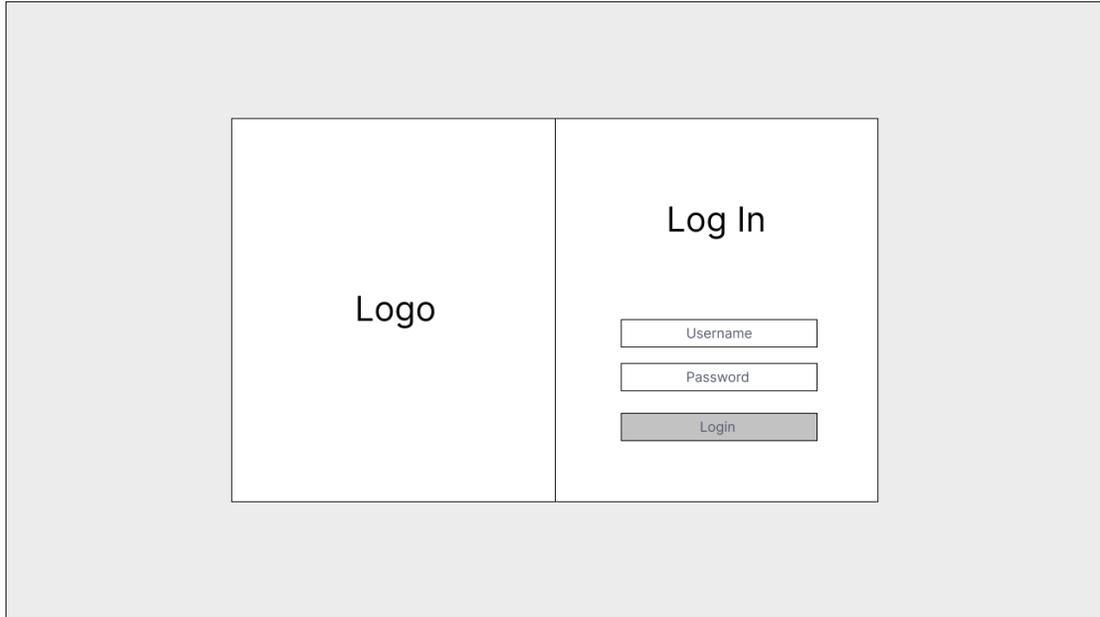
4.3.4 Entity Relationship Diagram



Gambar 4. 20 Class Diagram Helpdesk

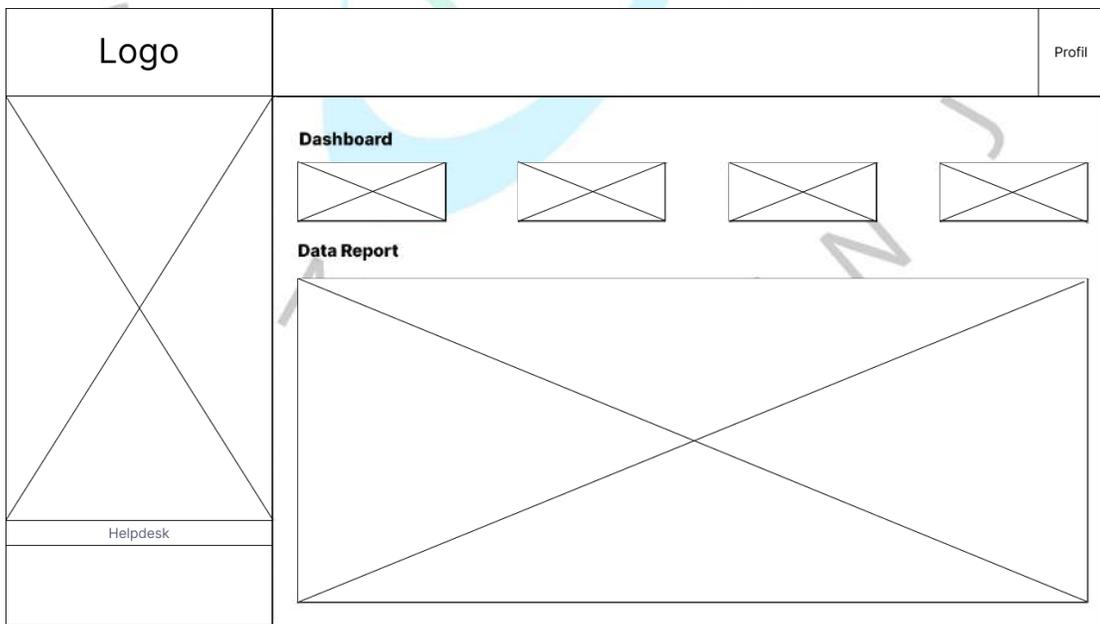
4.4 Perancangan Interface

Perancangan antarmuka adalah mengubah sistem menjadi bentuk yang lebih nyata. Proses ini melibatkan pembuatan *prototype* dan desain sistem itu sendiri yang akan dibangun. Berikut adalah perancangan antarmuka yang telah dibuat oleh peneliti.



Gambar 4. 21 Halaman *Login* MySAU

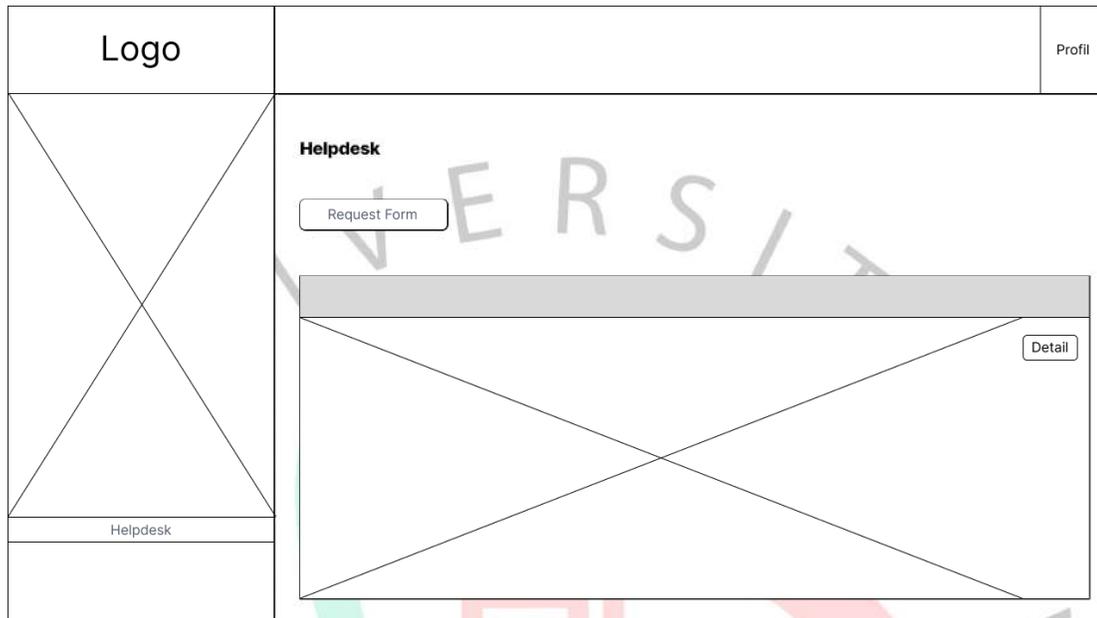
Pada gambar 4.19 merupakan rancangan antarmuka pada halaman login MySAU.



Gambar 4. 22 Halaman *Dashboard* MySAU

Pada gambar 4.20 merupakan rancangan antarmuka pada halaman *dashboard*. Terdapat grafik dari hasil data *report* dari data *request* yang dibuat oleh pekerja.

1) Perancangan Antarmuka *role* Pekerja



Gambar 4. 23 Halaman Menu *Helpdesk*

Pada gambar 4.21 merupakan rancangan antarmuka pada halaman menu *Helpdesk* dari *role* pekerja. Terdapat tabel pada menu *helpdesk* yang berisi data-data *request* dari pekerja.

Logo	Profil
Helpdesk	<p>Request Form</p> <div style="border: 1px solid black; padding: 5px;"> <p>Jenis Permintaan <input type="text" value="v"/></p> <p>Judul <input type="text"/></p> <p>Sampai Tanggal <input type="text"/></p> <p>Lampiran <input type="text" value="Pilih File"/></p> <p>Keterangan <input type="text"/></p> <p>Approval <input type="text" value="Remove"/> <input type="button" value="Remove"/></p> <p style="text-align: right;"><input type="button" value="Kembali"/> <input type="button" value="Send Request"/></p> </div>

Gambar 4. 24 Halaman *Request Form*

Pada gambar 4.22 merupakan rancangan antarmuka pada halaman *Request Form* dari *role* pekerja. Pada halaman *request form* digunakan untuk pekerja dapat mengajukan permintaan terkait permasalahan atau permintaan pada tim IT.

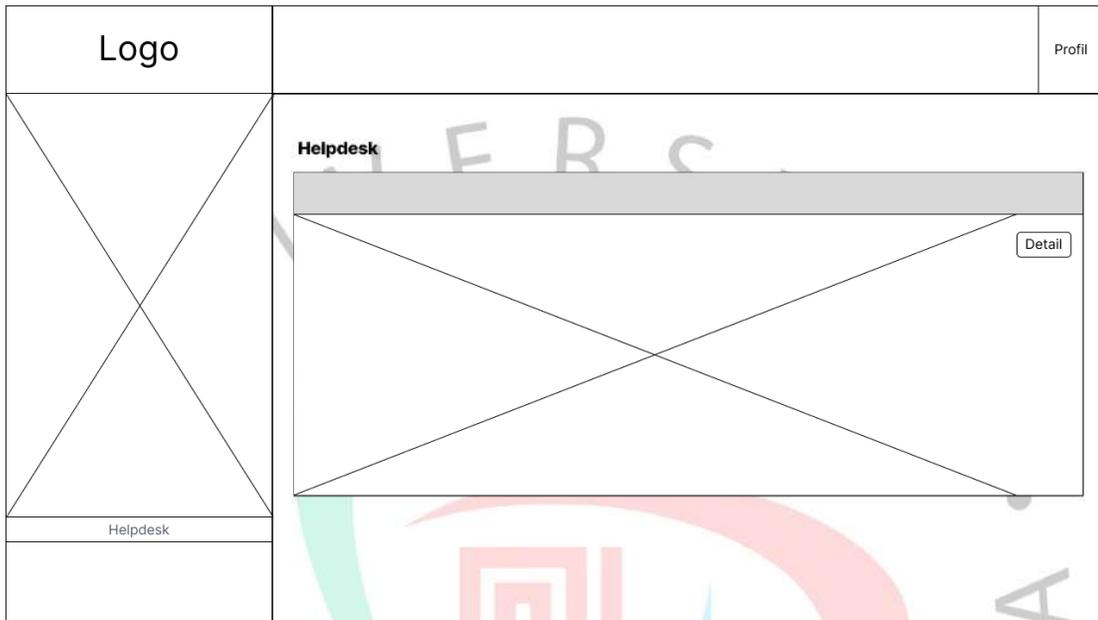
Logo	Profil
Helpdesk	<p>Detail Request</p> <div style="border: 1px solid black; padding: 5px; height: 150px;"> <p style="text-align: center; font-size: 2em;">X</p> </div> <div style="border: 1px solid black; padding: 5px; height: 150px;"> <p style="text-align: center; font-size: 2em;">X</p> </div> <p style="text-align: right;"><input type="button" value="Kembali"/> <input type="button" value="Edit"/> <input type="button" value="Batalkan"/></p>

Gambar 4. 25 Halaman *Detail Request*

Pada gambar 4.23 merupakan rancangan antarmuka pada halaman detail *request* dari *role* pekerja. Pada halaman detail *request* terdapat detail data *request* dan detail data pekerja yang telah mengajukan. Terdapat tombol edit untuk mengedit data *request*

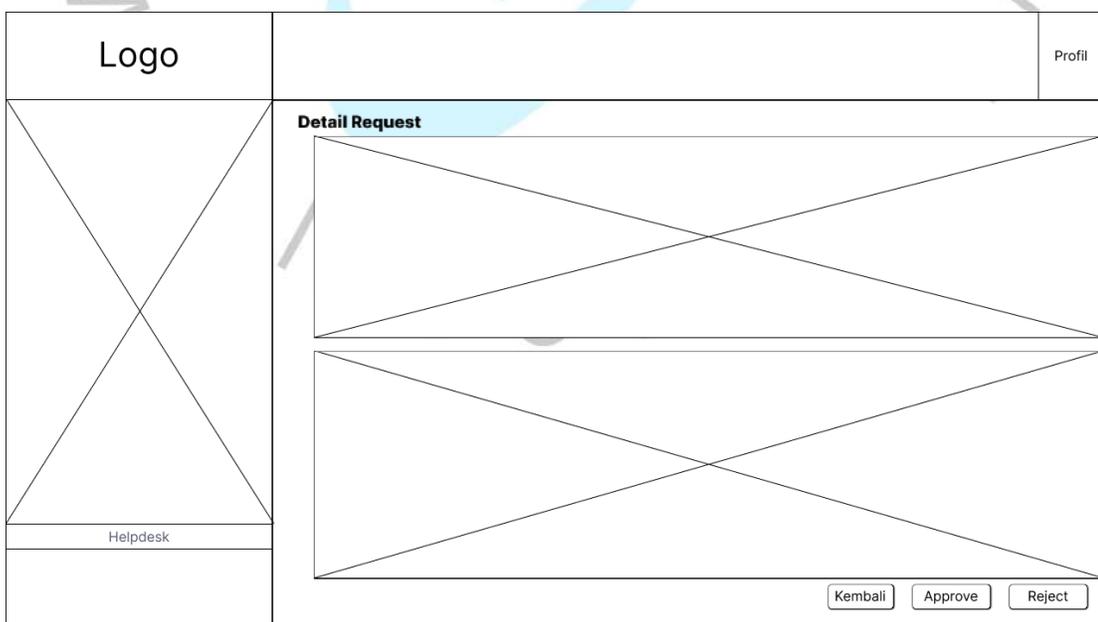
yang telah dibuat oleh pekerja, adapun tombol batalkan untuk pekerja yang ingin membatalkan data *request* yang telah diajukan.

2) Perancangan Antarmuka Manajemen



Gambar 4. 26 Halaman Menu *Helpdesk*

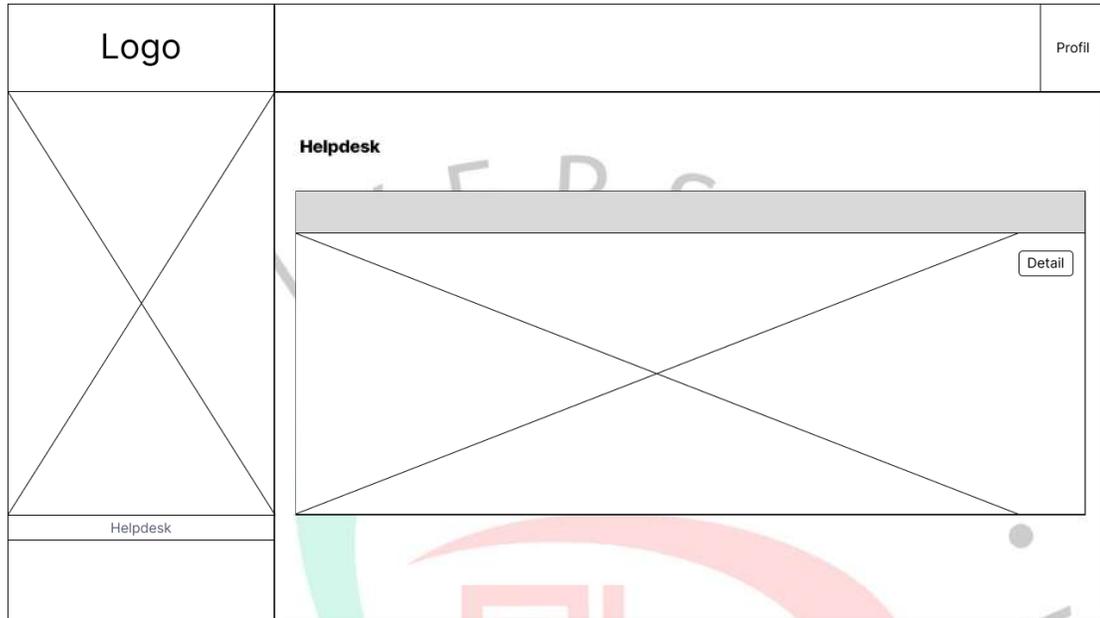
Pada gambar 4.24 merupakan rancangan antarmuka pada halaman menu *Helpdesk* dari *role* manajemen.



Gambar 4. 27 Halaman *Detail Request*

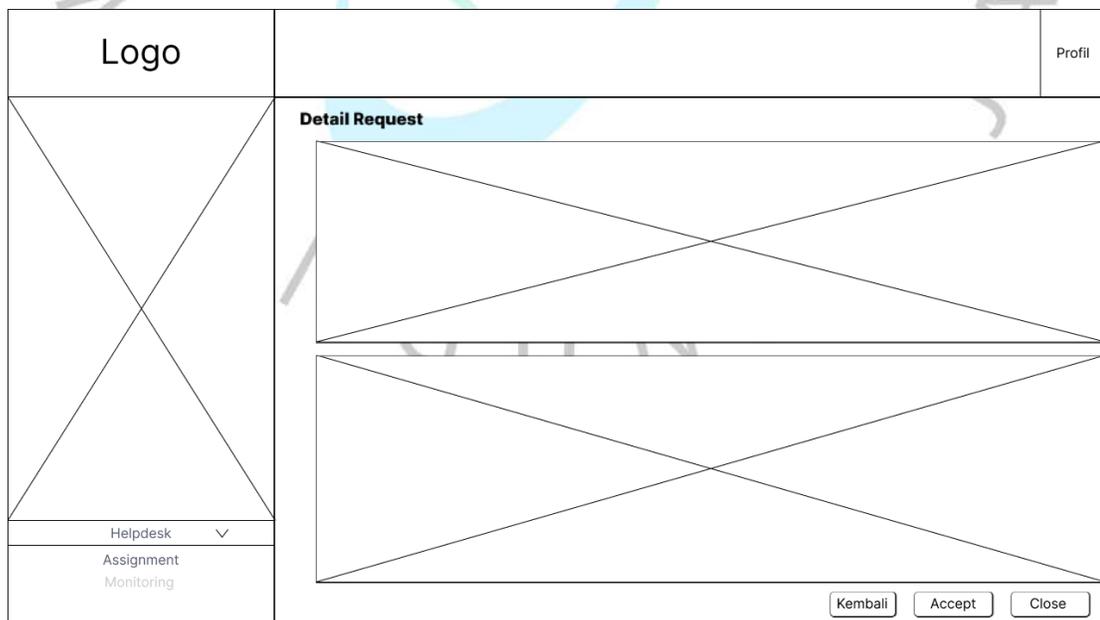
Pada gambar 4.25 merupakan rancangan antarmuka pada halaman detail *request* dari *role* manajemen. Pada halaman detail *request* terdapat tombol *approve* dan *reject*.

3) Perancangan Antarmuka Staf IT



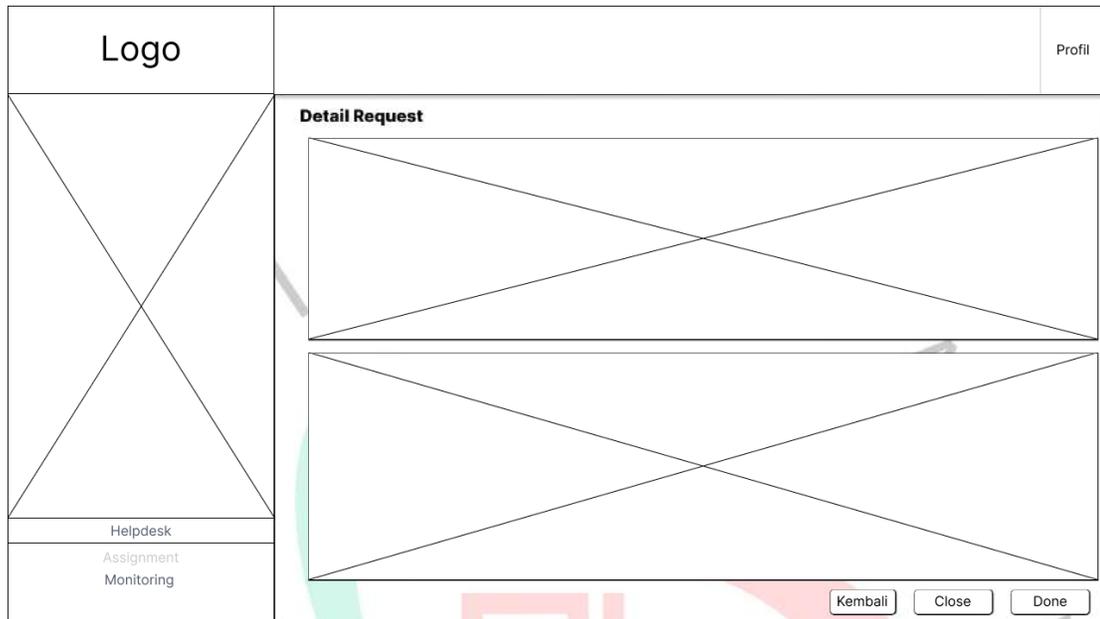
Gambar 4. 28 Halaman Menu *Assignment*

Pada gambar 4.26 merupakan rancangan antarmuka pada halaman menu *Helpdesk* di sub menu *assignment* dari *role* Staf IT.



Gambar 4. 29 Halaman Menu Detail *Assignment*

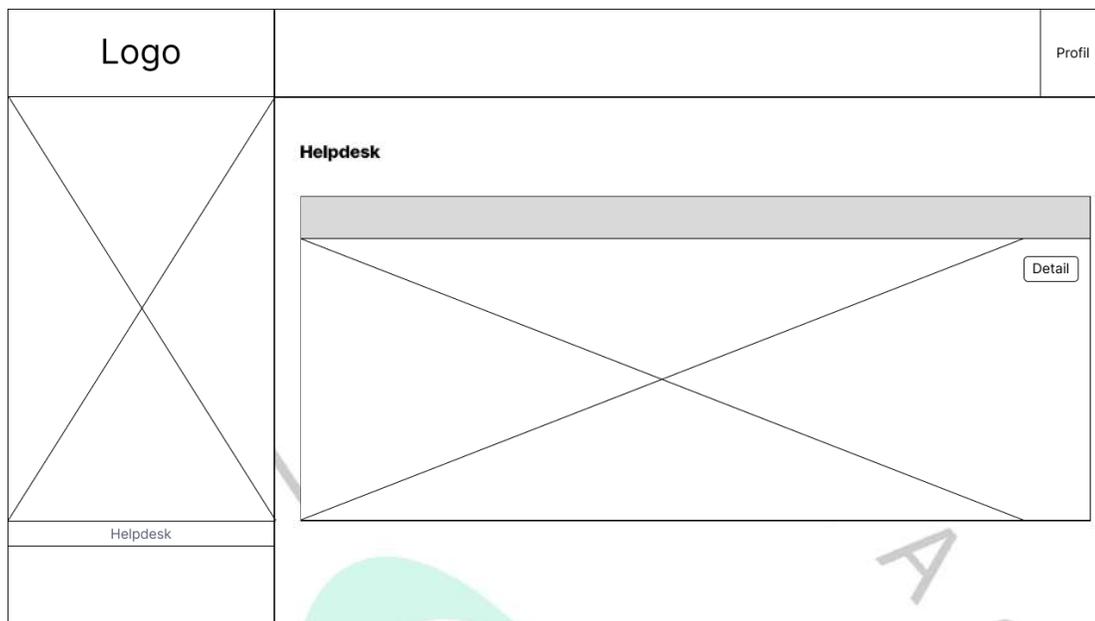
Pada gambar 4.27 merupakan rancangan antarmuka pada halaman detail *assignment* dari *role* staf IT. Pada halaman detail *assignment* terdapat tombol *accept* dan *Close*.



Gambar 4. 30 Halaman Menu Detail *Monitoring*

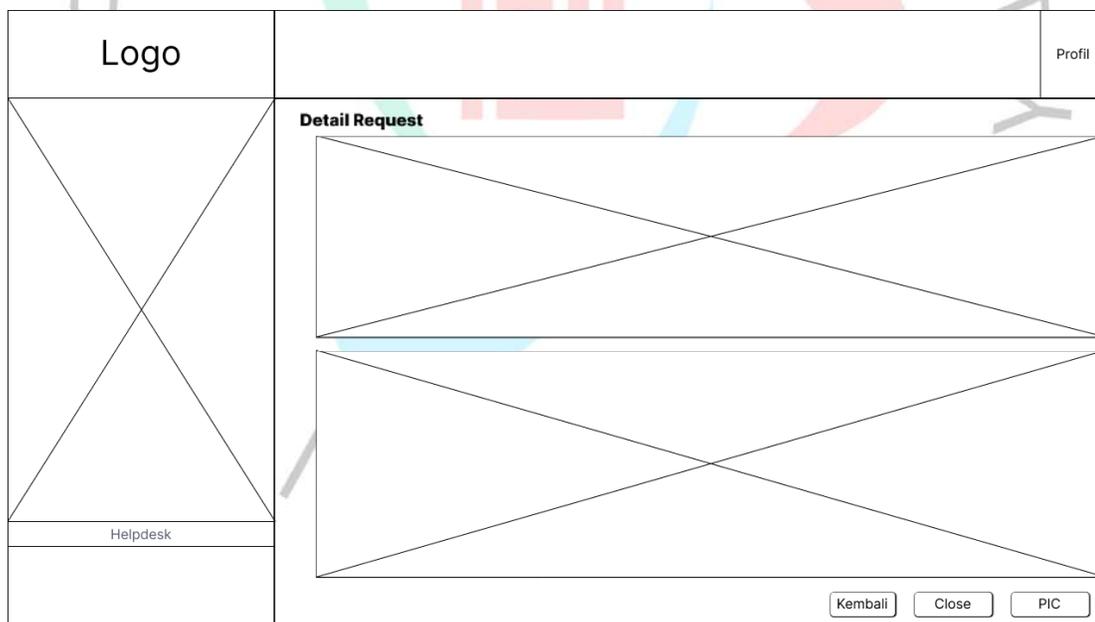
Pada gambar 4.28 merupakan rancangan antarmuka pada halaman detail *monitoring* dari *role* staf IT. Pada halaman detail *monitoring* terdapat tombol *Done* dan *Close*.

4) Perancangan Antarmuka Admin



Gambar 4. 31 Halaman Menu *Helpdesk*

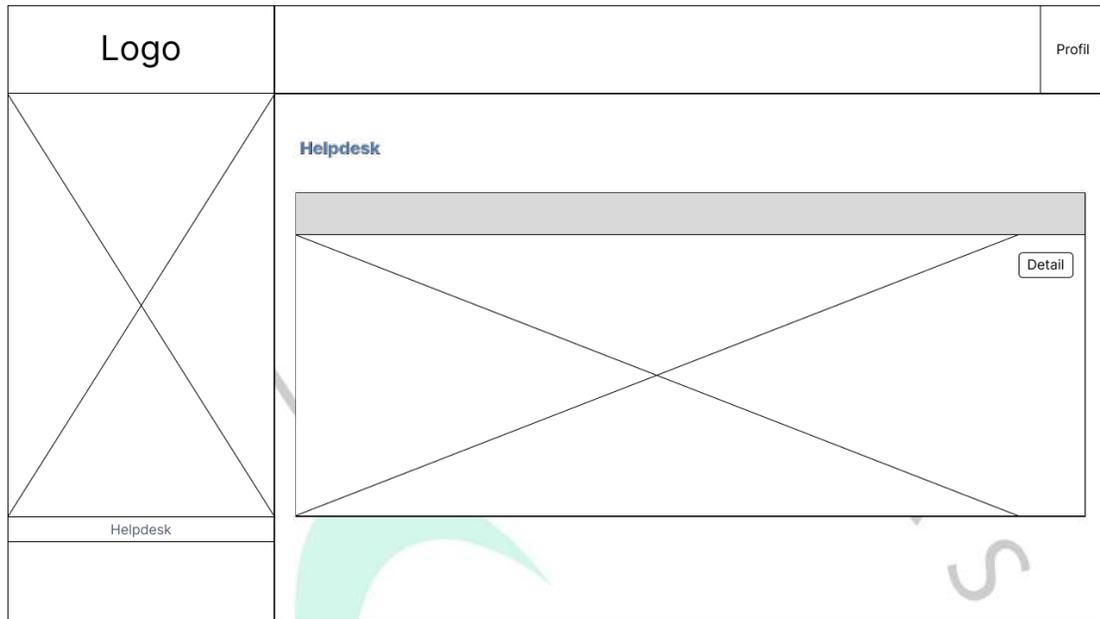
Pada gambar 4.29 merupakan rancangan antarmuka pada halaman menu *Helpdesk* dari *role* admin.



Gambar 4. 32 Halaman *Detail Request*

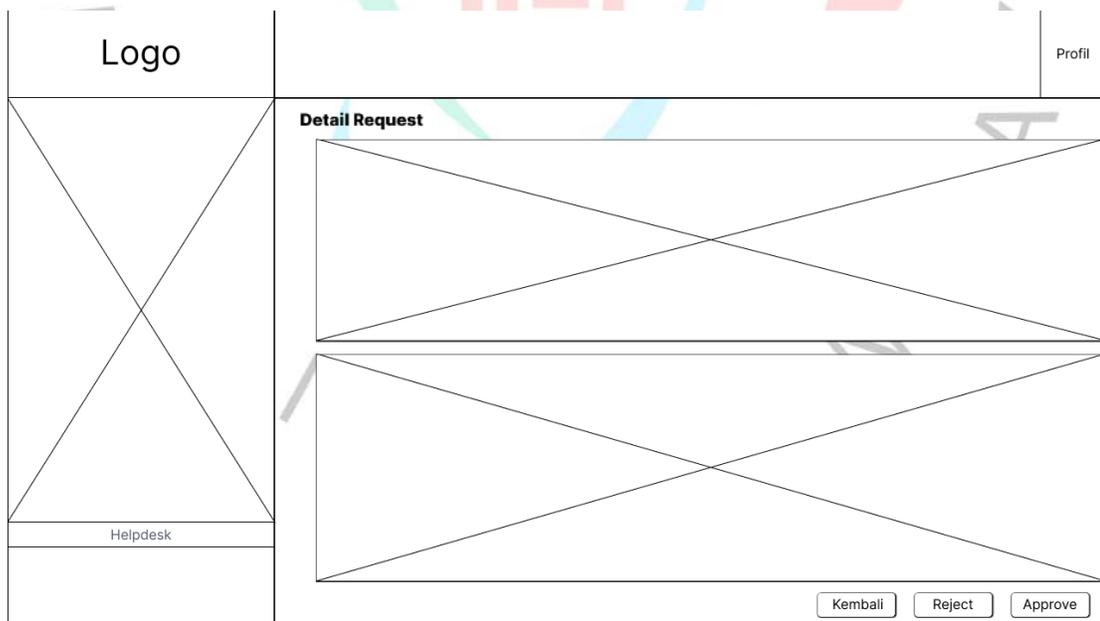
Pada gambar 4.30 merupakan rancangan antarmuka pada halaman detail *request* dari *role* admin. Pada halaman detail *request* di *role* admin terdapat tombol *PIC* dan *Close*.

5) Perancangan Antarmuka Manajemen IT



Gambar 4. 33 Halaman Menu *Helpdesk*

Pada gambar 4.31 merupakan rancangan antarmuka pada halaman menu *Helpdesk* dari *role* manajemen IT.



Gambar 4. 34 Halaman Detail *Request*

Pada gambar 4.32 merupakan rancangan antarmuka pada halaman detail *request* dari *role* manajemen IT.

4.5 Perancangan Black Box dan White Box

Pada sub bab ini, peneliti membuat perancangan *black box* dan *white box* pada aplikasi yang sedang dikembangkan menggunakan kedua metode tersebut sebagai berikut.

Tabel 4. 8 Perancangan Black Box

No.	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengamatan
1.	Tampilan Halaman <i>Login</i>	User melakukan login ke website MySAU dan masuk ke halaman <i>dashboard</i>	-
2.	Tampilan Halaman <i>Helpdesk</i>	User masuk ke menu <i>Helpdesk</i>	-
3.	Tampilan Halaman <i>Request Form</i>	Pekerja melakukan pengajuan permintaan pada halaman <i>Request Form</i> dan data dapat ditambahkan	-
4.	Tampilan Halaman Detail Request	Pekerja masuk ke halaman detail <i>request</i>	-
5.	Tampilan halaman Edit <i>Request</i>	Pekerja masuk ke halaman edit data <i>request</i>	-
6.	Tampilan halaman <i>Approve Request</i>	Manajemen masuk ke halaman detail dan melakukan <i>approve</i> data <i>request</i> . Data yang telah di <i>approve</i> akan hilang dari tabel manajemen	-
7.	Tampilan halaman <i>Monitoring</i>	Staf IT melakukan <i>done</i> data <i>request</i>	-
9.	Tampilan halaman Validasi data <i>Request</i>	Manajemen IT melakukan validasi data <i>request</i>	-
10.	Tampilan halaman Staf PIC	Admin memberikan data <i>request</i> ke staf IT pada projek yang belum diambil oleh Staf IT	-

Tabel 4. 9 Perancangan White Box

No	Komponen yang diuji	Kode Program	Hasil yang diharapkan
1	Halaman Login	<pre> <div class="col-md-8 col-lg-6 p-0 mx-auto"> <div class="bg-white text-dark br-7 br-tl-0 br-bl-0"> <div class="card-body"> <div class="text-center mb-3"> <h1 class="mb-2">Log In</h1> Hello There ! </div> <?php if (session()->getFlashdata('msg')): ?> <div class="alert alert-danger"> <? session()->getFlashdata('msg') ?> </div> <?php endif; ?> <form class="mt-5" method="POST" action="<? base_url('login/auth') ?>"> <div class="input-group mb-4"> <div class="input-group-text"> <i class="fa-solid fa-user"></i> </div> <input type="text" class="form-control" placeholder="Username" name="username" required> </div> <div class="input-group mb-4"> <div class="input-group id="Password-toggle"> <i class="fa-solid fa-lock" aria-hidden="true"></i> <input class="form-control" type="password" placeholder="Password" name="password" required> </div> </div> <div class="form-group"> <label class="custom-control custom-checkbox"> <input type="checkbox" class="custom-control-input" /> Remember Me </label> </div> <div class="form-group text-center mb-3"> <input type="submit" class="btn btn-lg w-100 br-7 text-white" style="background-color: #005197;" value="Login"> </div> <div class="form-group fs-13 text-center"> Forget Password ? </div> </form> </div> </div> </div> </pre>	Menampilkan halaman <i>login</i>
Hasil Pengamatan			

2	<p>Halaman Data Report</p>	<pre> <head> <h4 class="page-title mb-5 text-primary">Data Report</h4> <i class="fa fa-download me-2 fs-14"></i>Download <script src="https://cdn.jsdelivr.net/npm/chart.js"></script> </head> <body> <canvas id="myChart"></canvas> <script> // konfigurasi chart var config = { type: 'line', data: { labels: [<?php foreach (\$request_form as \$row) { echo "' . \$row['jenis_permintaan'] . ','; } ?>], datasets: [[label: 'Maret', data: [<?php foreach (\$request_form as \$row) { echo \$row['total'] . ','; } ?>], borderColor: 'rgb(255, 99, 132)', fill: false,]] } }; // buat chart baru var ctx = document.getElementById('myChart').getContext('2d'); var myChart = new Chart(ctx, config); </script> </body> </pre>	<p>Menampilkan grafik data report</p>
---	----------------------------	---	---------------------------------------

Hasil Pengamatan

3	<p>Halaman Helpdesk</p>	<pre> <?=\$this->extend('template/main') ?> <?=\$this->section('content') ?> <div class="page-header"> <div class="page-leftheader"> <h4 class="page-title mb-0 text-primary">Helpdesk</h4> </div> </div> <div class="page-header"> <div class="page-leftheader"> <?php if (session()->get('role') == '2') : ?> ? base_url('pengajuan/requestform') ?>" class="btn btn-primary">Request Form <?php endif; ?> </div> </div> <!-- Tabel Maker --> <table id="example" class="table table-bordered table-striped text-center"> <thead class="table-head"> <tr> <th width="5%">No</th> <th width="15%">Judul</th> <th width="23%">Jenis Permintaan</th> <th width="33%">keterangan</th> <th>Status</th> <?php if (session()->get('role') == '2') : ?> <th>SLA</th> <?php endif; ?> <th>Action</th> </tr> </thead> </pre>	<p>Menampilkan halaman Helpdesk</p>
---	-------------------------	--	-------------------------------------

```

<tbody>
  <?php
    $statuses = array(
      0 => 'Dalam Pengajuan',
      1 => 'Disetujui',
      2 => 'Ditolak',
      3 => 'On Progress',
      4 => 'Done',
      5 => 'Cancel',
      6 => 'Close',
      7 => 'Validasi'
    );
    $nomor = 1;
    foreach ($request_form as $value) { ??
      <tr>
        <td><?=$ nomor ?></td>
        <td><?=$ value['judul'] ?></td>
        <td><?=$ value['jenis_permintaan'] ?></td>
        <td><?=$ value['keterangan'] ?></td>
        <td><?=$ isset($statuses[$value['status']]) ? $statuses[$value['status']] : '' ?></td>
        <?php if (session()->get('role') == '2') {
          if ($value['status'] == 3) {
            foreach ($ale as $row) {
              if ($row['Jenis_Permintaan'] == $value['jenis_permintaan']) {
                <td><?=$row['SLA'] ?></td>
                <?php
                  |
                }
              } else {
                <td><?=$row['SLA'] ?></td>
              }
            }
          }
        }
        <td>
          <a href="<?=$ base_url('Presensi/detail/' . $value['id']) ?>" class="btn btn-primary">
            Detail
          </a>
        </td>
      </tr>
      <?php $nomor += 1;
    }
  }
  </tbody>
</table>

<script>
$(document).ready(function() {
  $('#filter-table').DataTable();
});
</script>

```

Hasil Pengamatan

4	<p>Halaman Request Form</p>	<pre> <?=\$this->extend('template/main') ?> <?=\$this->section('content') ?> <div class="page-header"> <div class="page-lefthead"> <h4 class="page-title mb-0 text-primary">Request Form</h4> </div> </div> <form action="{base_url('presensi/simpanRequest')}" method="POST" enctype="multipart/form-data" class="card row g-3"> <div class="col-md-11"> <label for="jenis_permintaan" class="form-label">Jenis Permintaan </label> <select class="form-select" id="jenis_permintaan" name="jenis_permintaan" aria-label="Default select example" required> <option value="" selected disabled>-- Pilih Jenis Permintaan --</option> <option value="Manage Service Aplikasi">Manage Service Aplikasi</option> <option value="Pembuatan Aplikasi">Pembuatan Aplikasi</option> <option value="Pembuatan Dokumen">Pembuatan Dokumen</option> <option value="Pembuatan Rencan/Prototype">Pembuatan Rencan/Prototype</option> <option value="Pengadaan Barang">Pengadaan Barang</option> <option value="Pengembangan Aplikasi">Pengembangan Aplikasi</option> <option value="Setting Jaringan Internet">Setting Jaringan Internet</option> <option value="Setting Server">Setting Server</option> <option value="Setting/Update Laptop">Setting/Update Laptop</option> </select> </div> <div class="col-md-11"> <label for="judul" class="form-label">Judul</label> <input type="text" name="judul" class="form-control" id="judul" required> </div> <div class="col-md-11"> <label for="upload" class="form-label"> Lampiran </label> <input type="file" name="lampiran" id="lampiran" class="form-control form-file" required /> </div> <div class="col-md-11"> <label for="keterangan" class="form-label">Keterangan </label> <textarea rows="6" name="keterangan" id="keterangan" placeholder="Type your message" class="form-control" required></textarea> </div> <form> <div class="row mt-5"> <div class="col-md-11"> <div id="inputFormRow"> <label class="form-label" for="approval">Approval</label> <div class="input-group mb-3"> <select type="text" name="approval[]" class="form-control" data-placeholder="Enter title" autocomplete="off" required> <optgroup> <option value="" selected disabled>-- Pilih Approval --</option> <option value="m1">Mas Septien</option> <option value="m2">Mas Balang</option> </optgroup> </select> <div class="input-group-append"> <button id="addRow" type="button" class="btn btn-info">+</button> </div> </div> </div> <div id="newRow"></div> </div> </div> <div class="d-grid gap-2 d-md-flex justify-content-md-end mb-4"> Kembali <button class="btn btn-primary" type="submit">Send Request</button> </div> </form> <script> \$(document).ready(function() { \$("#inputFormRow").show(); \$("#jenis_permintaan").on("change", function() { if (this.value == "Manage Service Aplikasi" this.value == "Pembuatan Aplikasi" this.value == "Pengembangan Aplikasi") { \$("#inputFormRow").show(); \$("#addRow").show(); } else { \$("#inputFormRow").hide(); \$("#addRow").hide(); } }); }); </script> </pre>	Menampilkan halaman request form
---	-----------------------------	--	----------------------------------

		<pre> \$("#addRow").click(function() { var html = ''; html += '<div class="input-group mb-3">'; html += '<select type="text" name="approval[]" class="form-control" data-placeholder="Enter title" autocomplete="off" required>'; html += '<optgroup>'; html += '<option value="" selected disabled-- Pilih Approval --</option>'; html += '<option value="1">Septian</option>'; html += '<option value="2">Galang</option>'; html += '</optgroup>'; html += '</select>'; html += '<div class="input-group-append">'; html += '<button id="removeRow" type="button" class="btn btn-danger"></button>'; html += '</div>'; html += '</div>'; \$('#newRow').append(html); }); \$(document).on('click', '#removeRow', function() { if (\$('#newRow').children().length > 0) { \$(this).closest('.input-group').remove(); } }); </script> <?= \$this->endSection() ?> </pre>	
--	--	--	--

Hasil Pengamatan

5	<p>Halaman Detail Request</p>	<pre> <?= \$this->extend('template/main') ?> <?= \$this->section('content') ?> <div class="page-header"> <div class="page-leftheader"> <h4 class="page-title mb-0 text-primary">Detail Request</h4> </div> </div> <div class="card"> <div class="card-body"> <table class="table table-borderless"> <thead> <tr> <th colspan="6"> <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0;"> <small>Statuses = array</small> <small>0 => 'Dalam Pengejuan',</small> <small>1 => 'Disetujui',</small> <small>2 => 'Ditolak',</small> <small>3 => 'On Progress',</small> <small>4 => 'Done',</small> <small>5 => 'Cancel',</small> <small>6 => 'Close'</small> </pre>
---	---------------------------------------	---


```

<script>
    $("#btn-show-sweetalert").click(function(cancel) {
        Swal.fire({
            title: 'Anda yakin ingin dibatalkan?',
            showDenyButton: true,
            confirmButtonText: 'Iya',
            denyButtonText: 'Tidak',
        }).then((result) => {
            /* Read more about isConfirmed, isDenied below */
            if (result.isConfirmed) {
                $.ajax({
                    url: '<?>= base_url('presensi/updateStatus') ?>',
                    type: "POST",
                    data: {
                        'id': <?>= $detail['id'] ?>,
                        'status': '5',
                    },
                    success: function(data, textStatus, jqXHR) {
                        Swal.fire('Saved!', '', 'success')
                        window.location.href = "<?>= base_url('presensi') ?>";
                    },
                    error: function(jqXHR, textStatus, errorThrown) {
                        alert('Error occurred!');
                    }
                });
            }
        });
    });

    $("#btn-show-sweetalert2").click(function(close) {
        Swal.fire({
            title: 'Anda yakin ingin di close?',
            showDenyButton: true,
            confirmButtonText: 'Iya',
            denyButtonText: 'Tidak',
        }).then((result) => {
            /* Read more about isConfirmed, isDenied below */
            if (result.isConfirmed) {
                $.ajax({
                    url: '<?>= base_url('presensi/closeRequest') ?>',
                    type: "POST",
                    data: {
                        'id': <?>= $detail['id'] ?>,
                        'status': '6',
                    },
                    success: function(data, textStatus, jqXHR) {
                        Swal.fire('Saved!', '', 'success')
                        window.location.href = "<?>= base_url('presensi') ?>";
                    },
                    error: function(jqXHR, textStatus, errorThrown) {
                        alert('Error occurred!');
                    }
                });
            }
        });
    });
</script>

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11.7.3/dist/sweetalert2.all.min.js"></script>
<link href="https://cdn.jsdelivr.net/npm/sweetalert2@11.7.3/dist/sweetalert2.min.css" rel="stylesheet">

<!-- <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script> -->

```

Hasil Pengamatan