

Sistem Informasi Pengelolaan Anggaran

PROYEK AKHIR

Disusun Oleh:

Firjatullah Anang Setiawan

3312311111

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

2026

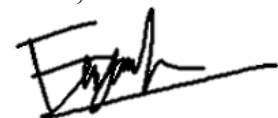
KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga laporan Proyek Akhir yang berjudul "Sistem Informasi Pengelolaan Anggaran" ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Ahli Madya Teknik Informatika pada Program Studi Teknik Informatika, Politeknik Negeri Batam. Dalam proses penyelesaiannya, penulis banyak mendapat dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu Alena Uperiati, S.T., M.Cs selaku dosen pembimbing akademik yang telah memberikan arahan dan masukan selama proses pengerjaan proyek ini.
2. Bapak Nur Cahyono Kushardianto, S.Si., M.T., M.Sc., Ph.D serta rekan dari Unit Transformasi Digital selaku pembimbing lapangan yang telah memberikan bimbingan teknis selama pelaksanaan proyek di Politeknik Negeri Batam.
3. Seluruh dosen dan staf Program Studi Teknik Informatika Politeknik Negeri Batam yang telah memberikan ilmu dan dukungan selama masa perkuliahan.
4. Semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyelesaian laporan ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang membutuhkan.

Batam, 12 Juni 2026



Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR TABEL	iv
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN.....	viii
ABSTRAK	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan	3
1.4. Batasan Masalah.....	3
1.5. Manfaat	3
1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Tinjauan Solusi dan Sistem Terkait	7
2.2. Landasan Konsep dan Teknologi	9
2.3. Metode Pengembangan Produk.....	12
BAB III ANALISIS DAN PERANCANGAN	15
3.1. Analisis Kebutuhan	15
3.2. Perancangan	16
BAB IV IMPLEMENTASI DAN PENGUJIAN	53
4.1. Hasil Implementasi.....	53
4.2. Pengujian.....	58
BAB V KESIMPULAN	73

5.1 Kesimpulan	73
5.2 Saran.....	73
DAFTAR PUSTAKA	75
DAFTAR LAMPIRAN.....	77

DAFTAR TABEL

Tabel 2. 1 Tinjauan Aplikasi Sistem Informasi Anggaran Sejenis	8
Tabel 3. 1 Kebutuhan Fungsional	17
Tabel 3. 2 Kebutuhan Non Fungsional	17
Tabel 3. 3 Skenario <i>Use case</i> Melakukan <i>Login</i>	21
Tabel 3. 4 Skenario <i>Use case</i> Melakukan <i>Logout</i>	21
Tabel 3. 5 Skenario <i>Use case</i> Membuat Pengajuan SPBy atau SPM	22
Tabel 3. 6 Skenario <i>Use case</i> Melihat Status Pengajuan SPBy atau SPM	22
Tabel 3. 7 Skenario <i>Use case</i> Mengelola Jadwal Anggaran dan Entry Progres Fisik.....	23
Tabel 3. 8 Skenario <i>Use case</i> Melakukan Verifikasi Pengajuan SPBy atau SPM LS	24
Tabel 3. 9 Skenario <i>Use case</i> Menyetujui atau Menolak Pengajuan SPBy atau SPM LS	25
Tabel 3. 10 Skenario <i>Use case</i> Memvalidasi dan Menyetujui Pengajuan yang Telah Diverifikasi.....	25
Tabel 3. 11 Skenario <i>Use case</i> Mencatat Entry Realisasi SPBy atau SPM.....	26
Tabel 3. 12 Skenario <i>Use case</i> Mencetak Dokumen Pengajuan yang Disetujui ..	27
Tabel 3. 13 Skenario <i>Use case</i> Mengelola Master Data Anggaran.....	27
Tabel 3. 14 Skenario <i>Use case</i> Melihat Semua Data Anggaran dan Realisasi	28
Tabel 3. 15 Skenario <i>Use case</i> Mengakses Laporan Kegiatan Anggaran.....	29
Tabel 3. 16 Skenario <i>Use case</i> Mengelola Mapping Pagu Unit dan PPK ke Akun	29
Tabel 3. 17 Skenario <i>Use case</i> Mengelola Tahun Anggaran	30
Tabel 3. 18 Skenario <i>Use case</i> Mengelola Unit dan User	31
Tabel 3. 19 Jadwal Penelitian	51
Tabel 4. 1 <i>Test case</i> pengguna melakukan <i>login</i>	59
Tabel 4. 2 <i>Test case</i> pengguna melakukan <i>logout</i>	60
Tabel 4. 3 <i>Test case</i> Staf membuat pengajuan SPBy atau SPM LS.....	60
Tabel 4. 4 <i>Test case</i> Staf melihat status pengajuan SPBy atau SPM LS	62

Tabel 4. 5 <i>Test case</i> Kepala Unit mengelola jadwal anggaran dan entry progres fisik.....	63
Tabel 4. 6 <i>Test case</i> Kepala Unit melakukan verifikasi pengajuan SPBy atau SPM LS	64
Tabel 4. 7 <i>Test case</i> menyetujui atau menolak pengajuan SPBy atau SPM LS....	65
Tabel 4. 8 <i>Test case</i> PPK memvalidasi dan menyetujui pengajuan yang telah diverifikasi.....	65
Tabel 4. 9 <i>Test case</i> Bendahara mencatat entry realisasi SPBy atau SPM	66
Tabel 4. 10 <i>Test case</i> mencetak dokumen pengajuan yang disetujui.....	67
Tabel 4. 11 <i>Test case</i> KPA mengelola master data anggaran	68
Tabel 4. 12 <i>Test case</i> KPA melihat semua data anggaran dan realisasi	69
Tabel 4. 13 <i>Test case</i> KPA mengakses laporan kegiatan anggaran	70
Tabel 4. 14 <i>Test case</i> KPA mengelola mapping pagu unit dan PPK ke akun.....	70
Tabel 4. 15 <i>Test case</i> KPA mengelola tahun anggaran.....	71
Tabel 4. 16 <i>Test case</i> KPA mengelola unit dan user.....	72

DAFTAR GAMBAR

Gambar 3. 1 Gambaran Umum Sistem	16
Gambar 3. 2 Diagram <i>Flowchart</i> SIPA	18
Gambar 3. 3 <i>Use case Diagram</i>	20
Gambar 3. 4 <i>Activity diagram</i> Melakukan <i>Login</i>	32
Gambar 3. 5 <i>Activity diagram</i> Melakukan <i>Logout</i>	32
Gambar 3. 6 <i>Activity diagram</i> Membuat Pengajuan SPBy atau SPM	33
Gambar 3. 7 <i>Activity diagram</i> Melihat Status Pengajuan SPBy atau SPM	33
Gambar 3. 8 <i>Activity diagram</i> Mengelola Jadwal Anggaran dan Entry Progres Fisik	34
Gambar 3. 9 <i>Activity diagram</i> Melakukan Verifikasi terhadap Pengajuan SPBy atau SPM	35
Gambar 3. 10 <i>Activity diagram</i> Menyetujui atau Menolak Pengajuan SPBy atau SPM	36
Gambar 3. 11 <i>Activity diagram</i> Memvalidasi dan Menyetujui Pengajuan yang telah Diverifikasi	37
Gambar 3. 12 <i>Activity diagram</i> Mencatat Entry Realisasi SPBy atau SPM	38
Gambar 3. 13 <i>Activity diagram</i> Mencetak Dokumen Pengajuan yang Disetujui .	39
Gambar 3. 14 <i>Activity diagram</i> Mengelola Master Data Anggaran	40
Gambar 3. 15 <i>Activity diagram</i> Melihat Semua Data Anggaran dan Realisasi	41
Gambar 3. 16 <i>Activity diagram</i> Mengakses Laporan Kegiatan Anggaran	42
Gambar 3. 17 <i>Activity diagram</i> Mengelola Mapping Pagu Unit dan PPK ke Akun	43
Gambar 3. 18 <i>Activity diagram</i> Mengelola Tahun Anggaran	44
Gambar 3. 19 <i>Activity diagram</i> Mengelola Unit dan User	45
Gambar 3. 20 <i>Entity relationship diagram</i> SIPA	46
Gambar 3. 21 <i>Relational Schema</i>	47
Gambar 3. 22 <i>Wireframe Login</i>	48
Gambar 3. 23 <i>Wireframe</i> Pengajuan SPBy atau SPM	49
Gambar 3. 24 <i>Wireframe</i> Formulir Pengajuan Baru	49
Gambar 3. 25 <i>Wireframe</i> Tambah Item	50

Gambar 3. 26 <i>Wireframe</i> Entry Realisasi SPBy GU atau SPM LS	50
Gambar 3. 27 <i>Wireframe</i> Laporan Kegiatan Anggaran	51
Gambar 4. 1 Tampilan <i>Login</i>	53
Gambar 4. 2 Tampilan <i>Dashboard</i>	54
Gambar 4. 3 Tampilan Halaman Pengajuan SPBy	54
Gambar 4. 4 Tampilan Halaman Pengajuan SPM LS.....	54
Gambar 4. 5 Tampilan Form Pengajuan SPBy	55
Gambar 4. 6 Tampilan Form Pengajuan SPM LS	55
Gambar 4. 7 Tampilan Halaman Entry Realisasi SPBy.....	55
Gambar 4. 8 Tampilan Halaman Entry Realisasi SPM LS	56
Gambar 4. 9 Tampilan Halaman Verifikasi Pengajuan SPBy atau SPM LS	56
Gambar 4. 10 Tampilan Master Data Anggaran	57
Gambar 4. 11 Tampilan Riwayat Data Pengajuan SPBy	57
Gambar 4. 12 Tampilan Riwayat Data Pengajuan SPM LS	57
Gambar 4. 13 Tampilan Laporan Kegiatan Anggaran	58

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement	77
Lampiran B. Link Produk	77
Lampiran C. Dokumen Pengujian UAT.....	77

ABSTRAK

Perkembangan teknologi informasi mendorong perguruan tinggi untuk memiliki sistem administrasi yang efektif, efisien, dan terintegrasi guna mendukung tata kelola keuangan. Politeknik Negeri Batam selama ini menggunakan Sistem Informasi Pengelolaan Anggaran berbasis C#, namun sistem tersebut menghadapi berbagai keterbatasan, seperti sering terjadinya error, ketergantungan pada pengembang *eksternal*, dan kesulitan integrasi dengan aplikasi *internal*. Penelitian ini bertujuan untuk melakukan migrasi total SIPA ke platform PHP dengan framework Laravel, mencakup proses perencanaan pagu anggaran, pengajuan, verifikasi, realisasi, hingga pelaporan anggaran yang dikenal memiliki arsitektur terstruktur, keamanan bawaan, serta dukungan komunitas yang luas. Fokus utama penelitian meliputi perancangan ulang sistem, migrasi data, serta pengujian integrasi dengan aplikasi *internal* Polibatam. Hasil yang diharapkan adalah sistem informasi anggaran yang lebih stabil, terintegrasi, mudah dipelihara, serta dapat dikelola secara mandiri oleh tim *internal*, sehingga mendukung peningkatan transparansi dan efektivitas tata kelola keuangan institusi.

Kata Kunci: Laravel, Migrasi Sistem, Pengelolaan Anggaran, PHP, Politeknik Negeri Batam, Sistem Informasi.

BAB I PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi yang semakin pesat menuntut perguruan tinggi untuk memiliki sistem administrasi yang efektif, efisien, dan terintegrasi guna mendukung tata kelola organisasi secara optimal. Dalam konteks tersebut, sistem informasi tidak hanya berperan dalam mempercepat proses pengolahan data, tetapi juga menjadi instrumen penting dalam mendukung pengambilan keputusan strategis serta meningkatkan transparansi pengelolaan keuangan kampus. Oleh karena itu, penerapan sistem informasi yang andal menjadi kebutuhan utama bagi institusi pendidikan tinggi dalam menghadapi tuntutan pengelolaan administrasi yang semakin kompleks (Effendi, n.d.).

Politeknik Negeri Batam selama ini menggunakan Sistem Informasi Pengelolaan Anggaran berbasis C# sebagai sarana utama dalam mendukung proses perencanaan, pengajuan, hingga pelaporan anggaran. Namun, meskipun sistem tersebut telah berfungsi, berbagai kendala teknis dan operasional masih sering terjadi, seperti frekuensi error yang mengganggu akurasi data, ketergantungan terhadap pengembang *eksternal*, serta kesulitan dalam integrasi dengan aplikasi *internal* lainnya. Kondisi tersebut berpotensi menimbulkan hambatan pada proses perencanaan, pengelolaan, dan pelaporan anggaran secara menyeluruh. Permasalahan ini sejalan dengan penelitian mengenai *modernisasi* sistem *legacy* yang menunjukkan bahwa migrasi sistem sering menghadapi tantangan berupa kompatibilitas data, kontinuitas layanan, serta kesiapan sumber daya manusia dalam beradaptasi dengan sistem baru (Hogan *et al.*, 2024).

Sebagai upaya untuk mengatasi berbagai kendala tersebut, migrasi sistem ke platform PHP dengan framework Laravel menjadi salah satu solusi yang layak dipertimbangkan. Laravel dinilai memiliki struktur pengembangan yang lebih *modern*, keterbacaan kode yang lebih baik, serta dukungan API yang memudahkan integrasi dengan berbagai sistem *internal* maupun *eksternal*. Selain itu, berbagai penelitian terkait implementasi Laravel dalam pengembangan sistem informasi pendidikan menunjukkan bahwa framework ini mampu meningkatkan *maintainability*, efisiensi pengembangan, dan produktivitas pengembang.

Meskipun demikian, proses implementasinya tetap memerlukan perhatian khusus pada aspek optimasi performa, pengelolaan basis data, dan perencanaan migrasi yang matang agar sistem dapat berjalan secara optimal (Ariyanto *et al.*, 2024).

Keputusan untuk melakukan migrasi total dan tidak melanjutkan pengembangan pada platform C# (ASP.NET) yang sudah ada didasarkan pada beberapa pertimbangan strategis. Pertama, pengembangan pada sistem lama selama ini bergantung pada vendor *eksternal*, sehingga setiap tahun institusi perlu menghubungi dan membayar pihak ketiga untuk pemeliharaan maupun pengembangan fitur. Migrasi ke Laravel memungkinkan sistem dikembangkan dan dipelihara secara mandiri (*in-house*) oleh tim *internal*. Kedua, migrasi ini juga menjadi bagian dari upaya standarisasi teknologi sistem informasi di lingkungan Politeknik Negeri Batam, mengingat perbedaan tumpukan teknologi (*technology stack*) antar sistem *internal* akan menyulitkan proses pemeliharaan jangka panjang. Ketiga, dari sisi infrastruktur, sistem berbasis C# sebelumnya memerlukan dua server terpisah untuk operasionalnya, sedangkan pada sistem baru kebutuhan tersebut disederhanakan menjadi satu server sehingga lebih efisien dari segi biaya dan pengelolaan. Keempat, dari perspektif keberlanjutan jangka panjang, Laravel dipilih karena kemudahan penulisan kode dan kurva belajar yang relatif lebih rendah dibandingkan ASP.NET, sehingga pengembangan fitur baru di masa mendatang dapat dilakukan secara lebih cepat dan mudah oleh tim *internal*.

Berdasarkan kondisi operasional yang dihadapi serta dukungan dari berbagai kajian literatur, penelitian ini difokuskan pada migrasi total SIPA dari platform C# ke PHP menggunakan framework Laravel. Tujuan utama dari penelitian ini adalah menghasilkan sistem yang lebih stabil, mudah dipelihara, terintegrasi dengan kebutuhan institusi, serta mampu mengurangi ketergantungan terhadap pihak pengembang *eksternal*. Selain itu, penelitian ini juga akan memperhatikan strategi mitigasi risiko migrasi, termasuk proses pemindahan data historis, pengujian kontinuitas layanan, dan peningkatan kapabilitas sumber daya manusia *internal* agar proses transisi dapat berjalan secara aman, efektif, dan berkelanjutan.

1.2. Rumusan Masalah

Rumusan masalah yang dibahas dalam penelitian ini adalah:

1. Bagaimana merancang dan membangun ulang Sistem Informasi Pengelolaan Anggaran menggunakan Framework Laravel?
2. Bagaimana menjaga integritas data dan keberlanjutan operasional dari aplikasi yang sudah ada?
3. Bagaimana melakukan pengujian sistem menggunakan metode *User Acceptance Test (UAT)* untuk memastikan kesesuaian fungsi sistem dengan kebutuhan pengguna?

1.3. Tujuan

Penelitian ini bertujuan untuk:

1. Merancang dan membangun ulang Sistem Informasi Pengelolaan Anggaran menggunakan Framework Laravel.
2. Menjaga integritas data serta memastikan keberlanjutan operasional sistem melalui strategi migrasi yang aman dan terstruktur.
3. Melakukan pengujian sistem menggunakan metode *User Acceptance Test (UAT)* untuk memastikan bahwa sistem yang dikembangkan telah sesuai dengan kebutuhan dan alur kerja pengguna di lingkungan Polibatam.

1.4. Batasan Masalah

Untuk menjaga fokus penelitian, batasan masalah penelitian ini adalah:

1. Penelitian hanya berfokus pada migrasi Sistem Informasi Pengelolaan Anggaran dari platform C# ke PHP menggunakan framework Laravel.
2. Sistem yang dikembangkan mencakup modul utama, yaitu perencanaan, pengajuan, dan pelaporan anggaran.
3. Integrasi yang dibahas terbatas pada aplikasi *internal* Polibatam, tidak mencakup sistem *eksternal* di luar institusi.

1.5. Manfaat

Proyek akhir ini diharapkan dapat memberikan kontribusi nyata baik secara keilmuan maupun aplikatif, dengan rincian sebagai berikut:

1. Manfaat Teoritis (Ilmiah)

Secara akademis, penelitian ini diharapkan dapat memberikan manfaat berupa: Referensi Migrasi Sistem: Menjadi studi literatur bagi peneliti atau mahasiswa lain mengenai metodologi migrasi sistem informasi dari teknologi *legacy* (C#) ke kerangka kerja *modern* (PHP Laravel). Pembuktian Implementasi Framework: Menyediakan bukti empiris mengenai efektivitas penggunaan framework Laravel dalam meningkatkan kemudahan pemeliharaan dan integrasi sistem di lingkungan institusi pendidikan.

2. Manfaat Praktis

Secara operasional, proyek ini memberikan dampak langsung bagi pihak-pihak terkait: Bagi Politeknik Negeri Batam: Menawarkan sistem informasi anggaran yang lebih stabil, cepat, dan terintegrasi dengan aplikasi *internal* lainnya. Ini meningkatkan efisiensi tata kelola keuangan kampus. Bagi Unit TI atau Pengembang *Internal*: Mengurangi ketergantungan pada pengembang *eksternal* dan meningkatkan kemandirian tim *internal* dalam melakukan kustomisasi serta pemeliharaan sistem di masa mendatang. Bagi Pengguna (*User*): Memudahkan proses perencanaan, pengajuan, dan pelaporan anggaran melalui antarmuka web yang lebih responsif dan minim kesalahan teknis.

3. Kontribusi terhadap SDGs

Proyek migrasi sistem ini secara spesifik mendukung pencapaian Tujuan Pembangunan Berkelanjutan (SDGs), khususnya pada:

- SDG No. 9: Industri, Inovasi, dan Infrastruktur: Proyek ini berkontribusi dalam meningkatkan infrastruktur teknologi di institusi pendidikan melalui inovasi digital. Dengan memodernisasi sistem *legacy* menjadi sistem berbasis *open-source* yang lebih tangguh, institusi memperkuat kapabilitas teknologi informasi yang berkelanjutan dan responsif terhadap perkembangan zaman.
- SDG No. 4: Pendidikan Berkualitas: Secara tidak langsung, pengelolaan anggaran yang lebih transparan dan efisien melalui sistem

yang andal akan mendukung penyediaan sarana dan prasarana pendidikan yang lebih baik di lingkungan Politeknik Negeri Batam.

1.6. Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Proyek Akhir ini dilaksanakan secara individu, di mana seluruh proses pengembangan sistem mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga dokumentasi dikerjakan oleh penulis sendiri. Meskipun demikian, pelaksanaan proyek tetap melibatkan kolaborasi dan sinergi dengan berbagai pihak untuk memastikan kualitas dan relevansi solusi yang dihasilkan. Selama pelaksanaan proyek, kolaborasi dilakukan dengan berbagai pihak sebagai berikut:

1. Diskusi rutin dengan pembimbing lapangan, Bapak Nur Cahyono Kushardianto, S.Si., M.T., M.Sc., Ph.D di Politeknik Negeri Batam, dilakukan secara berkala untuk memantau progres pengembangan serta mendapatkan arahan teknis terkait implementasi SIPA di lingkungan institusi.
2. Koordinasi dengan pihak *client*, yaitu unit terkait di Politeknik Negeri Batam, dilakukan dalam hal penyesuaian kebutuhan sistem. Salah satu bentuk penyesuaian yang dilakukan adalah penggantian modul autentikasi dari sistem *login* konvensional menjadi *Single Sign-On* (SSO) kampus, guna memastikan sistem terintegrasi dengan infrastruktur yang telah ada. Modul SSO tersebut bukan dikembangkan oleh penulis, melainkan disediakan oleh pihak institusi sebagai komponen integrasi.
3. Konsultasi dengan dosen pembimbing akademik, Ibu Alena Uperiati, S.T., M.Cs, dilakukan secara berkala untuk memastikan kesesuaian metodologi pengembangan sistem, alur migrasi, dan kelengkapan dokumentasi dengan standar akademik yang berlaku.
4. Pengumpulan umpan balik dari calon pengguna dilakukan melalui diskusi dengan staf yang terlibat dalam proses perencanaan dan pengelolaan anggaran di Politeknik Negeri Batam, serta melalui pengujian langsung menggunakan metode *User Acceptance Test* (UAT) untuk memastikan sistem sesuai dengan kebutuhan dan alur kerja operasional.

Bentuk kolaborasi ini menunjukkan bahwa meskipun proyek dilaksanakan secara individu, terdapat sinergi yang kuat dengan berbagai pemangku kepentingan. Keterlibatan aktif dari pembimbing lapangan, pihak *client*, dosen pembimbing, dan calon pengguna menjadi bagian penting dalam memastikan kualitas dan keberhasilan proyek migrasi SIPA ini.

BAB II TINJAUAN PUSTAKA

2.1. Tinjauan Solusi dan Sistem Terkait

Bagian ini membahas beberapa solusi dan sistem yang memiliki keterkaitan dengan pengembangan Sistem Informasi Pengelolaan Anggaran. Tinjauan dilakukan untuk memahami fitur yang telah diterapkan, teknologi yang digunakan, serta kelebihan dan keterbatasan dari sistem yang sudah ada sebagai bahan pertimbangan dalam pengembangan sistem pada proyek ini.

Sistem Informasi Perencanaan dan Realisasi Anggaran Dana Desa yang dikembangkan oleh (Suryani *et al.*, 2024) merupakan sistem yang dirancang untuk membantu pengelolaan dan penyampaian informasi realisasi anggaran desa kepada masyarakat. Sistem tersebut menggunakan *Flowchart*, *Data Flow Diagram* (DFD), *Entity relationship diagram* (ERD), serta perancangan antarmuka menggunakan Figma. Evaluasi dilakukan menggunakan metode *System Usability Scale* (SUS) dengan hasil *usability* yang baik. Namun, sistem masih berada pada tahap perancangan dan belum mencakup implementasi penuh.

Sistem Informasi Transparansi Anggaran dan Pendanaan (St Anna) berbasis web yang dikembangkan oleh (Purba *et al.*, 2025) dirancang untuk memfasilitasi manajemen keuangan *internal* kampus. Sistem ini berfokus pada peningkatan akuntabilitas pelaporan dan efisiensi proses validasi draf anggaran. Dengan memodelkan aliran data dan basis data relasional, sistem ini sukses memangkas birokrasi manual dan mempermudah pengawasan arus kas. Namun, cakupan integrasi sistem ini masih terbatas pada unit kerja *internal* tertentu dan belum terhubung ke sistem *eksternal* berskala besar.

Penerapan Sistem Anggaran berbasis web oleh (Saputra *et al.*, 2024) di BAPPEDA Kantor Walikota Medan berfokus pada migrasi pengelolaan anggaran dari metode konvensional ke platform digital. Sistem ini dirancang untuk mempercepat penyusunan laporan kerja instansi dan meminimalkan hambatan birokrasi. Hasilnya menunjukkan efisiensi waktu yang signifikan dalam rekapitulasi data, meskipun kendala teknis pada kestabilan server dan adaptasi pengguna awal masih menjadi catatan yang perlu dioptimalkan.

Sistem Informasi Akuntansi Perencanaan Anggaran Biaya Konstruksi yang dirancang oleh (Octavian Endra Saputra & Dendy Kurniawan, 2024) menerapkan metode *Activity Based Budgeting*. Sistem ini difokuskan untuk menghasilkan perencanaan biaya yang terstruktur guna menghindari salah input data dan mencegah deviasi anggaran proyek. Melalui akuntansi biaya yang sistematis, platform ini berhasil meningkatkan ketepatan estimasi, walaupun fleksibilitas fitur dalam menangani fluktuasi harga material di lapangan masih terbatas.

Untuk memperjelas perbandingan, penelitian-penelitian tersebut dirangkum dalam Tabel 2.1.

Tabel 2. 1 Tinjauan Aplikasi Sistem Informasi Anggaran Sejenis

Pengembang/ Tahun	Nama Aplikasi	Fitur Utama	Teknologi	Kelebihan	Keterbatasan
(Suryani et al., 2024)	Sistem Informasi Perencanaan dan Realisasi Anggaran Dana Desa	Pengelolaan anggaran, penyampaian informasi realisasi anggaran	<i>Flowchart</i> , DFD, ERD, Figma	Skor <i>usability</i> tinggi, mudah diakses masyarakat	Baru tahap desain, belum implementasi penuh
(Purba et al., 2025)	Sistem Informasi Transparansi Anggaran dan Pendanaan (St Anna)	Manajemen keuangan <i>internal</i> kampus, validasi draf anggaran, dan pelaporan	Web-based, Data Flow, Relational Database	Memangkas birokrasi manual dan mempermudah pengawasan arus kas	Integrasi masih terbatas pada unit <i>internal</i> tertentu, belum berskala besar
(Saputra et al., 2024)	Sistem Anggaran Berbasis Web pada BAPPEDA Walikota Medan	Migrasi data konvensional, percepatan penyusunan laporan kerja instansi	Web-based System	Efisiensi waktu yang signifikan dalam proses rekapitulasi data anggaran	Menghadapi kendala teknis pada kestabilan server dan adaptasi pengguna awal
(Octavian Endra Saputra & Dendy Kurniawan, 2024)	Sistem Informasi Akuntansi Perencanaan Anggaran Biaya Konstruksi	Perencanaan biaya terstruktur, kontrol input data, estimasi anggaran proyek	Web-based, <i>Activity Based Budgeting</i> (ABB)	Meningkatkan ketepatan estimasi biaya dan mencegah deviasi anggaran	Fitur masih terbatas dalam menangani fluktuasi harga material secara dinamis
Penelitian ini	Sistem Informasi Pengelolaan Anggaran	Pengajuan, validasi, pelaporan, integrasi sistem <i>internal</i>	PHP Laravel, PostgreSQL	Terstruktur, integrasi dengan aplikasi <i>internal</i> , pemeliharaan mandiri oleh	Tantangan migrasi data historis dan adaptasi SDM <i>internal</i>

				tim <i>internal</i>	
--	--	--	--	---------------------	--

Berdasarkan tinjauan solusi dan sistem terkait, sebagian besar sistem yang telah dikembangkan mampu membantu proses pengelolaan dan pelaporan anggaran secara digital. Namun, masih terdapat beberapa keterbatasan seperti implementasi yang belum menyeluruh, integrasi sistem yang terbatas, serta aspek keamanan dan keberlanjutan pengelolaan sistem.

Proyek ini berfokus pada pengembangan Sistem Informasi Pengelolaan Anggaran berbasis PHP Laravel dan PostgreSQL sebagai hasil migrasi dari sistem sebelumnya yang menggunakan C#. Sistem dikembangkan untuk mendukung proses pengajuan, validasi, dan pelaporan anggaran dalam satu platform terintegrasi sehingga diharapkan dapat meningkatkan efisiensi pengelolaan anggaran serta mempermudah pemeliharaan sistem oleh tim *internal*.

2.2. Landasan Konsep dan Teknologi

2.2.1. Sistem Informasi

Sistem informasi merupakan kumpulan elemen yang saling terkait untuk mengumpulkan, mengelola, menyimpan, dan menyebarkan informasi guna mendukung kelancaran aktivitas organisasi. Dalam konteks pengelolaan anggaran, sistem informasi berperan penting dalam menyusun, merencanakan, memonitor, dan melaporkan anggaran secara terintegrasi guna meningkatkan efisiensi dan transparansi. Proses perencanaan anggaran sendiri meliputi beberapa tahap utama seperti perencanaan, pengesahan, pelaksanaan, pelaporan, dan pertanggungjawaban yang harus didukung oleh sistem yang mampu memberikan data yang akurat dan tepat waktu. Hal ini sejalan dengan yang dikemukakan oleh (Irawan & Devitra, 2020), bahwa sistem informasi harus mampu mendukung seluruh siklus pengelolaan anggaran secara terintegrasi.

2.2.2. Perencanaan Anggaran

Perencanaan anggaran merupakan salah satu bagian krusial dalam manajemen organisasi yang berfungsi sebagai landasan strategis dan administratif dalam penggunaan sumber daya. (Suherdiansyah & Devitra, 2020) menjelaskan bahwa penggunaan teknologi informasi dalam perencanaan anggaran membantu

mengatasi masalah manual yang selama ini menyebabkan lambatnya proses pengajuan dan penyusunan laporan anggaran, serta ketidaksesuaian realisasi dengan rencana. Lebih lanjut, (Suherdiansyah & Devitra, 2020) juga menyatakan bahwa Sistem Informasi Pengelolaan Anggaran berbasis web memberikan kemudahan akses, *monitoring real-time*, dan kemampuan integrasi data sehingga mendukung pengambilan keputusan yang cepat dan tepat.

2.2.3. Sistem Informasi Pengelolaan Anggaran

Sistem Informasi Pengelolaan Anggaran (SIPA) di Politeknik Negeri Batam digunakan untuk mendukung seluruh siklus pengelolaan anggaran, mulai dari tahap perencanaan hingga pelaporan. Pada tahap perencanaan, Kuasa Pengguna Anggaran (KPA) menetapkan struktur Master Data Anggaran mengikuti hierarki RKAKL (Program, Kegiatan, Output, Suboutput, Komponen, Subkomponen, Akun, dan Item), serta menetapkan alokasi pagu untuk setiap unit kerja dan Pejabat Pembuat Komitmen (PPK) melalui fitur mapping pagu. Penetapan pagu ini dilakukan secara top-down oleh KPA dan mengikuti urutan bertahap, yaitu penetapan Master Data Anggaran terlebih dahulu, dilanjutkan dengan penetapan pagu per unit, dan terakhir pagu per PPK. Struktur dan alokasi inilah yang menjadi rencana anggaran, yang selanjutnya menjadi acuan bagi unit kerja dalam mengajukan Surat Perintah Bayar (SPBy) atau Surat Perintah Membayar Langsung (SPM LS).

Setelah rencana anggaran ditetapkan, unit kerja dapat mengajukan permintaan anggaran berdasarkan pagu yang tersedia. Setiap usulan kemudian diverifikasi oleh bagian terkait untuk memastikan kesesuaian dengan prioritas dan pagu anggaran yang tersedia. Jika diperlukan, usulan dapat direvisi sebelum memperoleh persetujuan akhir dari pihak berwenang.

Setelah disetujui, anggaran digunakan sebagai dasar pelaksanaan kegiatan dan realisasi dana. SIPA juga mendukung monitoring antara rencana dan realisasi anggaran sehingga memudahkan evaluasi, pengendalian, serta penyusunan laporan pertanggungjawaban. Dengan sistem ini, proses pengelolaan anggaran menjadi lebih terstruktur, transparan, dan efisien. Migrasi SIPA ke platform PHP

Laravel dilakukan untuk meningkatkan fleksibilitas pengembangan dan kemudahan integrasi sistem internal institusi.

2.2.4. Teknologi dan Tools yang Digunakan

1. PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman sisi server (*server-side scripting*) yang digunakan untuk membangun aplikasi berbasis web. PHP digunakan dalam proyek ini sebagai dasar pengembangan SIPA karena memiliki fleksibilitas tinggi dan dukungan komunitas yang luas (*PHP: Preface - Manual*, n.d.).

2. Laravel

Laravel merupakan framework PHP yang menerapkan arsitektur *Model-View-Controller* (MVC). Laravel digunakan dalam pengembangan SIPA untuk membantu pengelolaan kode program agar lebih terstruktur, meningkatkan keamanan aplikasi, serta mempermudah proses pengembangan dan pemeliharaan sistem (*Installation | Laravel 13.x - The Clean Stack for Artisans and Agents*, n.d.).

3. PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional (*Relational Database Management System*) yang bersifat *open source*. PostgreSQL digunakan dalam proyek ini untuk menyimpan data anggaran, data pengguna, serta informasi lain yang diperlukan sistem. PostgreSQL dipilih karena memiliki kemampuan pengelolaan data yang baik, stabilitas tinggi, serta mendukung kebutuhan sistem berskala besar (*PostgreSQL: Documentation*, n.d.).

4. Arsitektur MVC

MVC merupakan pola arsitektur perangkat lunak yang membagi aplikasi menjadi tiga komponen utama yaitu *Model*, *View*, dan *Controller*. Penerapan MVC membantu meningkatkan keteraturan kode program sehingga proses pengembangan dan pemeliharaan aplikasi menjadi lebih mudah (*MVC (Model, View, Controller) - BINUS @Bekasi - Kampus Beken Asyik | Business Service and Technology*, n.d.).

2.2.5. Profil Lokasi Penerapan Solusi

Politeknik Negeri Batam merupakan institusi pendidikan vokasi yang berfokus pada pengembangan kompetensi berbasis industri dan teknologi. Dalam aktivitas operasional institusi, proses perencanaan dan pengelolaan anggaran menjadi bagian penting untuk mendukung pelaksanaan program kerja dan kegiatan akademik maupun non-akademik.

Sebelumnya, Sistem Informasi Pengelolaan Anggaran menggunakan platform berbasis C# sehingga terdapat kebutuhan untuk meningkatkan fleksibilitas pengembangan sistem, kemudahan pemeliharaan, serta integrasi dengan aplikasi *internal* lainnya. Oleh karena itu, dilakukan pengembangan dan migrasi SIPA ke platform berbasis PHP Laravel dan PostgreSQL.

Pengembangan sistem ini diharapkan dapat membantu proses pengajuan, validasi, *monitoring*, dan pelaporan anggaran agar lebih efektif, efisien, serta mendukung pengelolaan anggaran yang lebih baik di lingkungan institusi.

2.3. Metode Pengembangan Produk

Dalam proyek migrasi Sistem Informasi Pengelolaan Anggaran ini, metode pengembangan perangkat lunak yang dipilih adalah *Waterfall*. Metode ini dipilih karena karakteristiknya yang sistematis dan berurutan sangat sesuai untuk sistem dengan kebutuhan awal yang sudah jelas dan stabil seperti migrasi sistem *legacy* SIPA. (Sallu *et al.*, 2023) dalam penelitiannya yang dipublikasikan di Jurnal Teknologi Pendidikan menyatakan bahwa metode *Waterfall* adalah model siklus hidup pengembangan perangkat lunak yang mengalir secara linear dan berurutan, dengan fokus pada penyelesaian satu tahap sebelum melanjutkan ke tahap berikutnya. Metode ini mendukung dokumentasi yang baik dan memudahkan pengendalian proyek yang terstruktur.

Tahapan pada metode *Waterfall* meliputi:

1. Analisis Kebutuhan

Tahapan ini bertujuan mengidentifikasi dan mendokumentasikan kebutuhan pengguna secara detail dan lengkap. (Sallu *et al.*, 2023) menegaskan bahwa tahap ini menjadi fondasi utama sebelum pengembangan sistem dimulai. Untuk itu, rencana kegiatannya adalah melaksanakan wawancara mendalam

dan studi dokumentasi dengan pengguna SIPA di Politeknik Negeri Batam guna mendapatkan spesifikasi kebutuhan sistem migrasi, termasuk kendala dalam penggunaan sistem lama berbasis C#.

2. Desain Sistem

Pada tahap ini dilakukan perancangan arsitektur sistem secara keseluruhan, termasuk desain basis data dan antarmuka pengguna. (Sallu *et al.*, 2023) menjelaskan bahwa desain sistem mempersiapkan *blueprint* teknis untuk fase implementasi. Sebagai realisasi dari kegiatan desain tersebut, akan didesain ulang struktur data dan modul aplikasi menggunakan framework Laravel, serta dirancang API untuk integrasi dengan aplikasi *internal* lainnya di kampus.

3. Implementasi

(Sallu *et al.*, 2023) mendefinisikan tahap ini sebagai proses penulisan kode program berdasarkan desain yang sudah disetujui. Kegiatan implementasi akan difokuskan pada pengkodean sistem migrasi SIPA ke PHP dengan framework Laravel secara bertahap sesuai modul yang telah dirancang.

4. Pengujian

(Sallu *et al.*, 2023) menyatakan bahwa pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Dalam pelaksanaannya, akan dilaksanakan pengujian unit, integrasi, sistem, serta *User Acceptance Test* (UAT) dengan melibatkan pengguna untuk melakukan uji coba, guna memastikan migrasi data berjalan lancar dan fungsionalitas sistem stabil.

5. Pemeliharaan

(Sallu *et al.*, 2023) menjelaskan bahwa tahap ini dilakukan setelah sistem dijalankan, untuk memperbaiki bug dan meningkatkan sistem sesuai umpan balik pengguna. Oleh karena itu, rencana kegiatan pasca-peluncuran meliputi penyusunan prosedur dukungan teknis dan *monitoring* kinerja sistem, serta mengadopsi umpan balik pengguna untuk pembaruan selanjutnya.

Metode *Waterfall* memungkinkan pengembangan sistem dilakukan secara terstruktur, terdokumentasi, dan terkontrol sehingga sangat sesuai untuk proyek migrasi SIPA yang memiliki kebutuhan sistem relatif jelas sejak awal. Dengan pendekatan ini, setiap tahapan dapat dilakukan secara sistematis guna

meminimalkan risiko kesalahan migrasi data, gangguan operasional, dan ketidaksesuaian kebutuhan sistem baru dengan proses bisnis yang telah berjalan, sebagaimana dikemukakan oleh (Sallu *et al.*, 2023).

BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

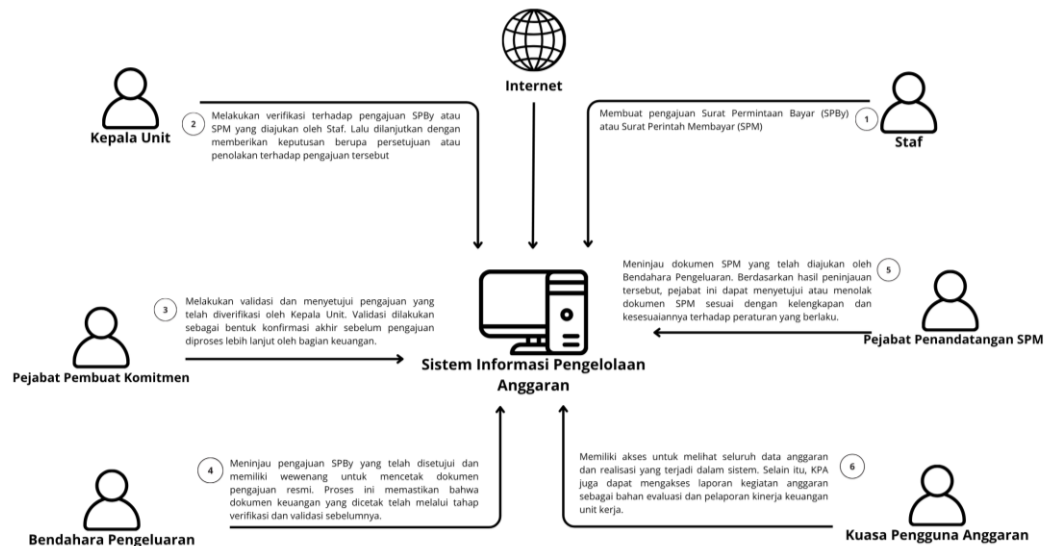
3.1.1 Proses Bisnis Berjalan

Pada proses bisnis yang berjalan sebelumnya, pengelolaan dokumen serta data anggaran di Politeknik Negeri Batam dilakukan melalui Sistem Informasi Pengelolaan Anggaran berbasis C#. Sistem ini digunakan untuk mendukung proses pencatatan, pengajuan, verifikasi, hingga pelaporan anggaran dalam kegiatan operasional institusi. Namun, dalam implementasinya masih terdapat berbagai kendala, seperti gangguan teknis yang memengaruhi keakuratan data, keterbatasan tim *internal* dalam melakukan pemeliharaan sistem secara mandiri, serta belum optimalnya integrasi dengan aplikasi *internal* lain di lingkungan Politeknik Negeri Batam.

Alur kerja pada sistem yang berjalan melibatkan beberapa aktor utama, yaitu Staf, Kepala Unit, Pejabat Pembuat Komitmen (PPK), Bendahara Pengeluaran, Pejabat Penandatangan Surat Perintah Membayar (SPM), dan Kuasa Pengguna Anggaran (KPA). Proses dimulai dari pengajuan dokumen anggaran oleh Staf, kemudian diverifikasi oleh Kepala Unit, dilanjutkan ke proses validasi oleh PPK dan Bendahara Pengeluaran atau Pejabat Penandatangan SPM, hingga tahap persetujuan akhir oleh Pejabat Pembuat Komitmen (PPK). Setiap tahapan dilakukan secara berurutan untuk memastikan dokumen anggaran, termasuk Surat Perintah Bayar (SPBy) dan Surat Perintah Membayar Langsung (SPM LS), telah sesuai dengan prosedur administratif yang berlaku.

Kondisi tersebut menjadi dasar dilaksanakannya pengembangan ulang sistem SIPA dengan arsitektur dan teknologi yang disesuaikan dengan kebutuhan pengelolaan anggaran di lingkungan Politeknik Negeri Batam. Pengembangan ulang ini diharapkan mampu mengatasi berbagai kendala teknis pada sistem sebelumnya, meningkatkan integrasi antar aplikasi *internal*, serta memperkuat efisiensi dalam proses pengajuan, verifikasi, dan pelaporan anggaran. Dengan demikian, sistem baru dapat memberikan dukungan operasional yang lebih stabil, fleksibel, dan mudah dipelihara oleh tim *internal*.

3.1.2 Gambaran Umum Sistem



Gambar 3. 1 Gambaran Umum Sistem

Sistem Informasi Pengelolaan Anggaran (SIPA) merupakan aplikasi berbasis web yang dibangun menggunakan framework Laravel dengan basis data PostgreSQL. Sistem ini digunakan untuk mengelola data dan dokumen anggaran yang meliputi tahapan perencanaan (penetapan master data anggaran dan pagu unit), pengajuan, verifikasi, realisasi, serta pelaporan dalam satu platform terpusat.

Setiap pengguna memiliki hak akses sesuai dengan peran dan tanggung jawabnya, meliputi Staf, Kepala Unit, Pejabat Pembuat Komitmen (PPK), Bendahara Pengeluaran, Pejabat Penandatangan Surat Perintah Membayar (PPSPM), dan Kuasa Pengguna Anggaran (KPA). Seluruh proses dilakukan secara berjenjang sesuai alur administrasi keuangan yang berlaku di Politeknik Negeri Batam.

3.2. Perancangan

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fungsi-fungsi utama yang harus disediakan oleh sistem berdasarkan interaksi antara pengguna (aktor) dan sistem pada Use case diagram. Kebutuhan fungsional ini terbagi dalam tiga tahap utama pengelolaan anggaran, yaitu tahap perencanaan, tahap pelaksanaan/realisasi, dan tahap pelaporan. Berikut ini merupakan kebutuhan fungsional Sistem Informasi Pengelolaan Anggaran:

Tabel 3. 1 Kebutuhan Fungsional

No	Kebutuhan Fungsional
FR-01	Melakukan <i>Login</i>
FR-02	Melakukan <i>Logout</i>
FR-03	Membuat Pengajuan SPBy atau SPM LS
FR-04	Melihat Status Pengajuan SPBy atau SPM LS sesuai Unit
FR-05	Mengelola Jadwal Anggaran dan Entry Progres Fisik
FR-06	Melakukan Verifikasi terhadap Pengajuan SPBy atau SPM LS
FR-07	Menyetujui atau Menolak Pengajuan SPBy atau SPM LS
FR-08	Memvalidasi dan Menyetujui Pengajuan yang telah Diverifikasi
FR-09	Mencatat Entry Realisasi SPBy atau SPM
FR-10	Mencetak Dokumen Pengajuan yang Disetujui
FR-11	Mengelola master data anggaran
FR-12	Melihat Semua Data Anggaran dan Realisasi
FR-13	Mengakses Laporan Kegiatan Anggaran
FR-14	Mengelola Mapping Pagu Unit dan PPK ke Akun
FR-15	Mengelola Tahun Anggaran
FR-16	Mengelola Unit dan User

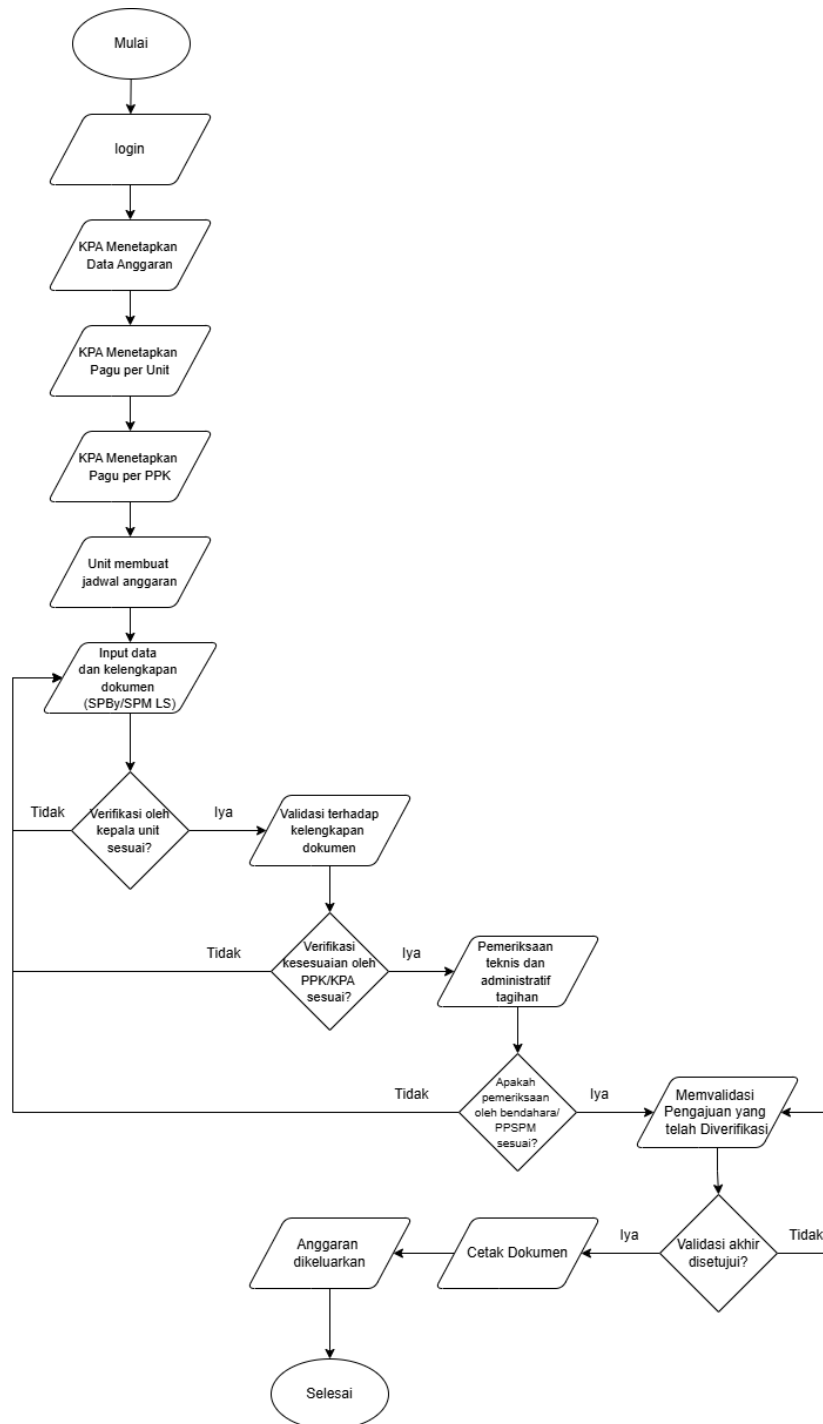
3.2.2 Kebutuhan Non-Fungsional

Berisi spesifikasi teknis yang tidak langsung terlihat dari fitur, tetapi penting untuk kualitas sistem.

Tabel 3. 2 Kebutuhan Non Fungsional

No	Parameter	Kebutuhan Non Fungsional
NFR-01	<i>Security</i>	Seluruh data pengguna, termasuk kredensial <i>login</i> dan transaksi pengajuan, harus dilindungi melalui enkripsi dan autentikasi yang aman. Akses ke sistem dibatasi berdasarkan hak pengguna (<i>role-based access control</i>).
NFR-02	<i>Usability</i>	Sistem memiliki antarmuka yang konsisten, mudah dipahami, dan dapat digunakan oleh pengguna dari berbagai level jabatan tanpa memerlukan pelatihan intensif.
NFR-03	<i>Performance</i>	Sistem harus mampu memberikan waktu respon yang cepat pada setiap proses seperti <i>login</i> , pengajuan, verifikasi, dan pelaporan, dengan waktu muat halaman tidak lebih dari 3 detik dalam kondisi normal.
NFR-04	<i>Scalability</i>	Sistem dirancang agar dapat dikembangkan lebih lanjut untuk menangani peningkatan jumlah data, pengguna, maupun fitur tambahan tanpa mengganggu kinerja yang sudah ada.

3.2.3 Diagram *Flowchart*



Gambar 3. 2 Diagram *Flowchart* SIPA

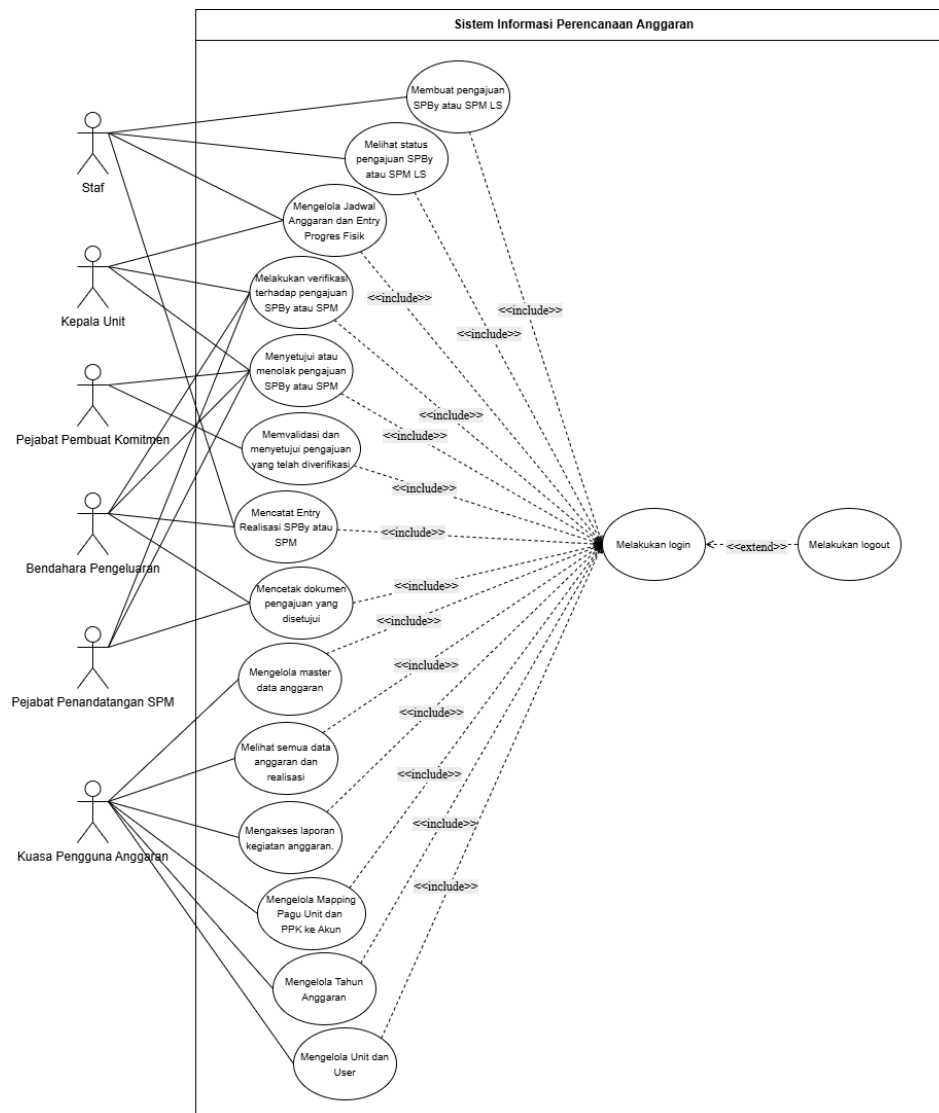
Sebelum proses pengajuan dapat dilakukan, KPA terlebih dahulu melaksanakan tahapan perencanaan anggaran. Berdasarkan flowchart tersebut, proses dimulai dari pengguna melakukan login ke dalam sistem. Selanjutnya,

KPA menetapkan Master Data Anggaran yang mengikuti struktur hierarki RKAKL (Program, Kegiatan, Output, Suboutput, Komponen, Subkomponen, Akun, dan Item). Berdasarkan Master Data Anggaran tersebut, KPA menetapkan pagu per unit, kemudian dilanjutkan dengan penetapan pagu per Pejabat Pembuat Komitmen (PPK) sebagai rincian dari pagu unit yang telah ditetapkan. Penentuan pagu ini bersifat top-down, yaitu KPA yang menentukan besaran pagu untuk masing-masing unit dan PPK.

Setelah pagu per PPK ditetapkan, unit kerja menyusun jadwal anggaran sebagai acuan waktu pelaksanaan kegiatan sesuai dengan pagu yang telah direncanakan. Selanjutnya, pengguna menginput data dan melengkapi dokumen pengajuan SPBy atau SPM LS sesuai pagu dan jadwal yang telah disusun. Setelah data dimasukkan, dilakukan verifikasi oleh kepala unit untuk memastikan kesesuaian pengajuan. Jika pengajuan belum sesuai, proses akan kembali ke tahap input data dan perbaikan dokumen. Namun, apabila telah sesuai, sistem akan melanjutkan ke tahap validasi kelengkapan dokumen dan verifikasi kesesuaian oleh PPK/KPA. Apabila pada tahap verifikasi ditemukan ketidaksesuaian, maka pengajuan akan dikembalikan untuk diperbaiki terlebih dahulu.

Setelah verifikasi dinyatakan sesuai, dilakukan pemeriksaan teknis dan administratif tagihan, kemudian dilanjutkan dengan pemeriksaan oleh bendahara/PPSPM untuk memastikan seluruh persyaratan telah terpenuhi. Jika hasil pemeriksaan belum sesuai, proses akan kembali ke tahap sebelumnya untuk dilakukan perbaikan. Apabila seluruh pemeriksaan telah sesuai, sistem akan melakukan validasi terhadap pengajuan yang telah diverifikasi dan dilanjutkan dengan validasi akhir. Jika validasi akhir disetujui, sistem akan mencetak dokumen, mengeluarkan anggaran, dan proses berakhir pada tahap selesai. Namun, apabila validasi akhir belum disetujui, pengajuan akan kembali ke tahap validasi untuk dilakukan peninjauan ulang.

3.2.4 Diagram *Use case*



Gambar 3. 3 *Use case Diagram*

Diagram *Use case* berikut menggambarkan interaksi antara pengguna dengan Sistem Informasi Pengelolaan Anggaran. Aktor yang terlibat meliputi Staf, Kepala Unit, Pejabat Pembuat Komitmen, Bendahara Pengeluaran, Pejabat Penandatangan SPM, dan Kuasa Pengguna Anggaran. Setiap aktor memiliki peran sesuai fungsinya, seperti melakukan *login*, membuat dan memverifikasi pengajuan SPBy atau SPM, meninjau serta menyetujui dokumen, hingga mengakses laporan kegiatan anggaran.

3.2.5 Skenario *Use case*

a. Skenario *Use case* Melakukan *Login*

Skenario pada tabel 3.3 dibawah ini menjelaskan proses autentikasi pengguna ke dalam sistem sebelum dapat mengakses fitur sesuai peran masing-masing.

Tabel 3. 3 Skenario *Use case* Melakukan *Login*

Skenario <i>Use case</i> 1: Melakukan <i>Login</i>	
Nomor	UC-01
Nama	Melakukan <i>login</i>
Deskripsi	Proses autentikasi pengguna sebelum mengakses sistem.
<i>Primary Actor</i>	Semua pengguna
<i>Preconditions</i>	Pengguna sudah memiliki akun terdaftar di sistem.
<i>Postconditions</i>	Pengguna berhasil <i>login</i> dan diarahkan ke <i>dashboard</i> sesuai peran.
<i>Main success scenario</i>	1. Pengguna membuka halaman <i>login</i>. 2. Sistem menampilkan form <i>login</i>. 3. Pengguna memasukkan <i>username</i> dan <i>password</i>. 4. Sistem memvalidasi data <i>login</i>. 5. Jika data valid, sistem mengarahkan pengguna ke <i>dashboard</i> masing-masing.
<i>Alternative Flow (Extension)</i>	4a. Jika <i>username</i> atau <i>password</i> salah, sistem menampilkan pesan “ <i>Login</i> gagal, periksa kembali kredensial Anda”.

b. Skenario *Use case* Melakukan *Logout*

Skenario pada tabel 3.4 dibawah ini menggambarkan proses pengguna keluar dari sistem untuk mengakhiri sesi dan menjaga keamanan akun.

Tabel 3. 4 Skenario *Use case* Melakukan *Logout*

Skenario <i>Use case</i> 2: Melakukan <i>Logout</i>	
Nomor	UC-02
Nama	Melakukan <i>logout</i>
Deskripsi	Pengguna keluar dari sistem untuk mengakhiri sesi penggunaan.
<i>Primary Actor</i>	Semua pengguna
<i>Preconditions</i>	Pengguna sudah <i>login</i> ke dalam sistem.
<i>Postconditions</i>	Sesi pengguna berakhir dan kembali ke halaman <i>login</i> .
<i>Main success scenario</i>	1. Pengguna menekan tombol <i>logout</i>. 2. Sistem menampilkan konfirmasi <i>logout</i>. 3. Pengguna mengonfirmasi <i>logout</i>. 4. Sistem menghapus sesi pengguna dan

	kembali ke halaman <i>login</i> .
<i>Alternative Flow (Extension)</i>	2a. Jika pengguna membatalkan <i>logout</i> , sistem tetap berada di halaman sebelumnya

c. Skenario *Use case* Membuat Pengajuan SPBy atau SPM

Skenario pada tabel 3.5 dibawah ini menjelaskan langkah staf dalam membuat dan mengirim pengajuan SPBy atau SPM ke sistem untuk diproses lebih lanjut.

Tabel 3. 5 Skenario *Use case* Membuat Pengajuan SPBy atau SPM

Skenario <i>Use case</i> 3: Membuat Pengajuan SPBy atau SPM	
Nomor	UC-03
Nama	Membuat Pengajuan SPBy atau SPM
Deskripsi	Staf membuat pengajuan SPBy atau SPM berdasarkan kebutuhan kegiatan.
<i>Primary Actor</i>	Staf
<i>Preconditions</i>	Staf sudah <i>login</i> dan memiliki hak akses untuk membuat pengajuan.
<i>Postconditions</i>	Pengajuan SPBy atau SPM tersimpan di sistem dengan status “Menunggu Verifikasi”.
<i>Main success scenario</i>	1. Staf membuka menu pengajuan. 2. Sistem menampilkan data ringkasan sesuai hak akses. 3. Staf mengisi data pengajuan. 4. Staf mengunggah dokumen pendukung 5. Staf menyimpan dan mengirim pengajuan. 6. Sistem menyimpan pengajuan dan mengubah status menjadi “Menunggu Verifikasi”.
<i>Alternative Flow (Extension)</i>	3a. Jika data tidak lengkap, sistem menampilkan pesan error “Lengkapi data sebelum mengirim pengajuan”.

d. Skenario *Use case* Melihat Status Pengajuan SPBy atau SPM

Skenario pada tabel 3.6 menjelaskan bagaimana staf memantau perkembangan status pengajuan yang telah dikirimkan. Staf membuka halaman status pengajuan, memilih salah satu pengajuan, lalu sistem menampilkan status terkini beserta catatan verifikasi dan riwayat prosesnya.

Tabel 3. 6 Skenario *Use case* Melihat Status Pengajuan SPBy atau SPM

Skenario <i>Use case</i> 4: Melihat Status Pengajuan SPBy atau SPM	
Nomor	UC-04
Nama	Melihat Status Pengajuan SPBy atau SPM
Deskripsi	Staf melihat perkembangan status pengajuan SPBy atau SPM yang telah diajukan.

<i>Primary Actor</i>	Staf
<i>Preconditions</i>	Staf sudah <i>login</i> dan pernah membuat pengajuan SPBy atau SPM.
<i>Postconditions</i>	Staf mendapatkan informasi status terkini dari pengajuan SPBy atau SPM.
<i>Main success scenario</i>	1. Staf membuka halaman status pengajuan. 2. Sistem menampilkan daftar pengajuan SPBy atau SPM milik staf. 3. Staf memilih salah satu pengajuan untuk melihat detailnya. 4. Sistem menampilkan status terbaru, catatan verifikasi, dan riwayat proses
<i>Alternative Flow (Extension)</i>	2a. Jika tidak ada pengajuan, sistem menampilkan pesan “Belum ada pengajuan yang dibuat”.

e. Skenario *Use case* Mengelola Jadwal Anggaran dan Entry Progres Fisik

Skenario pada tabel 3.7 menggambarkan proses Kepala Unit dalam mengatur jadwal pelaksanaan anggaran dan mencatat progres fisik kegiatan yang sedang berjalan. Kepala Unit memilih kegiatan, mengisi jadwal dan data progres.

Tabel 3. 7 Skenario *Use case* Mengelola Jadwal Anggaran dan Entry Progres Fisik

Skenario <i>Use case</i> 5: Mengelola Jadwal Anggaran dan Entry Progres Fisik	
Nomor	UC-05
Nama	Mengelola Jadwal Anggaran dan Entry Progres Fisik
Deskripsi	Kepala Unit mengelola jadwal pelaksanaan anggaran dan memasukkan data progres fisik kegiatan.
<i>Primary Actor</i>	Kepala Unit
<i>Preconditions</i>	Kepala Unit sudah <i>login</i> . Data anggaran tahun berjalan tersedia di sistem.
<i>Postconditions</i>	Jadwal anggaran dan data progres fisik berhasil disimpan di sistem.
<i>Main success scenario</i>	1. Kepala Unit memilih menu Jadwal Anggaran & Progres Fisik. 2. Sistem menampilkan daftar kegiatan anggaran unit 3. Kepala Unit memilih kegiatan yang akan dikelola. 4. Kepala Unit mengisi atau memperbarui jadwal pelaksanaan (tanggal mulai, tanggal selesai, target). 5. Kepala Unit memasukkan data progres fisik realisasi kegiatan. 6. Kepala Unit mengklik tombol Simpan. 7. Sistem memvalidasi dan menyimpan data jadwal serta progres fisik
<i>Alternative Flow (Extension)</i>	4a. Jika tanggal yang dimasukkan tidak valid, sistem menampilkan pesan kesalahan. 5a. Jika nilai progres fisik melebihi 100%, sistem

	menampilkan peringatan dan meminta konfirmasi.
--	--

f. Skenario *Use case* Melakukan Verifikasi Pengajuan SPBy atau SPM LS

Skenario pada tabel 3.8 menjelaskan proses Kepala Unit dalam memeriksa dan memverifikasi pengajuan yang masuk dari Staf. Kepala Unit membuka daftar pengajuan yang menunggu verifikasi, memeriksa kelengkapan data dan lampiran, lalu memilih untuk memverifikasi atau mengembalikan pengajuan.

Tabel 3. 8 Skenario *Use case* Melakukan Verifikasi Pengajuan SPBy atau SPM LS

Skenario <i>Use case</i> 6: Melakukan Verifikasi Pengajuan SPBy atau SPM LS	
Nomor	UC-06
Nama	Melakukan Verifikasi Pengajuan SPBy atau SPM LS
Deskripsi	Proses Kepala Unit melakukan verifikasi terhadap pengajuan SPBy atau SPM LS yang diajukan oleh Staf.
<i>Primary Actor</i>	Kepala Unit
<i>Preconditions</i>	Kepala Unit sudah <i>login</i> . Terdapat pengajuan dengan status "Menunggu Verifikasi".
<i>Postconditions</i>	Pengajuan berhasil diverifikasi dengan status berubah menjadi "Terverifikasi" atau "Dikembalikan".
<i>Main success scenario</i>	1. Kepala Unit memilih menu Verifikasi Pengajuan. 2. Sistem menampilkan daftar pengajuan yang menunggu verifikasi dari unitnya 3. Kepala Unit memilih pengajuan yang akan diverifikasi. 4. Sistem menampilkan detail pengajuan beserta lampiran. 5. Kepala Unit memeriksa kelengkapan dan kesesuaian data. 6. Kepala Unit memilih tombol Verifikasi. 7. Sistem memperbarui status menjadi "Terverifikasi"
<i>Alternative Flow (Extension)</i>	6a. Jika data tidak sesuai, Kepala Unit memilih tombol Kembalikan dan mengisi catatan alasan pengembalian. 6b. Sistem memperbarui status menjadi "Dikembalikan" dan mengirim notifikasi ke Staf.

g. Skenario *Use case* Menyetujui atau Menolak Pengajuan SPBy atau SPM LS

Skenario pada tabel 3.9 menggambarkan proses Pejabat Pembuat Komitmen (PPK), Bendahara, PPSPM dalam memberikan keputusan akhir terhadap pengajuan yang telah diverifikasi oleh Kepala Unit. Pejabat Pembuat Komitmen (PPK), Bendahara, PPSPM membuka daftar pengajuan terverifikasi, memeriksa detailnya, lalu memilih untuk menyetujui atau menolak disertai alasan jika ditolak.

Tabel 3. 9 Skenario *Use case* Menyetujui atau Menolak Pengajuan SPBy atau SPM LS

Skenario <i>Use case</i> 7: Menyetujui atau Menolak Pengajuan SPBy atau SPM LS	
Nomor	UC-07
Nama	Menyetujui atau Menolak Pengajuan SPBy atau SPM LS
Deskripsi	Proses menyetujui atau menolak pengajuan SPBy atau SPM LS yang telah diverifikasi.
<i>Primary Actor</i>	Pejabat Pembuat Komitmen (PPK), Bendahara, PPSPM
<i>Preconditions</i>	User sudah <i>login</i> . Terdapat pengajuan dengan status "Terverifikasi".
<i>Postconditions</i>	Status pengajuan berubah menjadi "Disetujui" atau "Ditolak" dan notifikasi dikirimkan.
<i>Main success scenario</i>	1. User memilih menu Persetujuan Pengajuan. 2. Sistem menampilkan daftar pengajuan yang telah terverifikasi. 3. User memilih dan memeriksa detail pengajuan. 4. User mengklik tombol Setujui. 5. Sistem memperbarui status menjadi "Disetujui".
<i>Alternative Flow (Extension)</i>	4a. Jika User menolak, memilih tombol Tolak dan mengisi alasan penolakan. 4b. Sistem memperbarui status menjadi "Ditolak".

h. Skenario *Use case* Memvalidasi dan Menyetujui Pengajuan yang Telah Diverifikasi

Skenario pada tabel 3.10 menjelaskan proses validasi akhir yang dilakukan oleh PPK terhadap pengajuan yang sudah disetujui. PPK memeriksa kelengkapan dokumen dan memberikan persetujuan final, sehingga pengajuan dapat diproses ke tahap pencairan.

Tabel 3. 10 Skenario *Use case* Memvalidasi dan Menyetujui Pengajuan yang Telah Diverifikasi

Skenario <i>Use case</i> 8: Memvalidasi dan Menyetujui Pengajuan yang Telah Diverifikasi	
Nomor	UC-08
Nama	Memvalidasi dan Menyetujui Pengajuan yang Telah

	Diverifikasi
Deskripsi	PPK memvalidasi dan menyetujui pengajuan yang telah diverifikasi
<i>Primary Actor</i>	PPK
<i>Preconditions</i>	PPK sudah <i>login</i> . Terdapat pengajuan dengan status "Menunggu persetujuan PPK".
<i>Postconditions</i>	Pengajuan dengan status "Disetujui".
<i>Main success scenario</i>	1. PPK memilih menu Entry Realisasi. 2. Sistem menampilkan daftar pengajuan. 3. PPK memilih dan memeriksa detail pengajuan. 4. PPK mengisi data diri dan mengklik tombol Setujui. 5. Sistem memperbarui status menjadi "Disetujui"
<i>Alternative Flow (Extension)</i>	-

i. Skenario *Use case* Mencatat Entry Realisasi SPBy atau SPM

Skenario pada tabel 3.11 menjelaskan proses Bendahara dalam mencatat realisasi pembayaran berdasarkan pengajuan yang telah disetujui. Bendahara memilih pengajuan, mengisi data realisasi, lalu menyimpannya sehingga saldo anggaran otomatis diperbarui oleh sistem.

Tabel 3. 11 Skenario *Use case* Mencatat Entry Realisasi SPBy atau SPM

Skenario <i>Use case</i> 9: Mencatat Entry Realisasi SPBy atau SPM	
Nomor	UC-09
Nama	Mencatat Entry Realisasi SPBy atau SPM
Deskripsi	Bendahara Pengeluaran mencatat realisasi pembayaran berdasarkan SPBy atau SPM yang telah disetujui.
<i>Primary Actor</i>	Bendahara Pengeluaran
<i>Preconditions</i>	Bendahara sudah <i>login</i> . Terdapat pengajuan dengan status "Disetujui".
<i>Postconditions</i>	Realisasi pembayaran berhasil dicatat dan saldo anggaran diperbarui.
<i>Main success scenario</i>	1. Bendahara memilih menu Entry Realisasi. 2. Sistem menampilkan daftar pengajuan yang siap direalisasikan 3. Bendahara memilih pengajuan dan mengisi data realisasi. 4. Bendahara mengklik tombol Simpan. 5. Sistem menyimpan realisasi, memperbarui saldo anggaran, dan mengubah status menjadi "Terealisasi".
<i>Alternative Flow (Extension)</i>	3a. Jika jumlah realisasi melebihi nilai pengajuan, sistem menampilkan peringatan. 4a. Jika data tidak lengkap, sistem menampilkan pesan error dan meminta koreksi.

j. Skenario *Use case* Mencetak Dokumen Pengajuan yang Disetujui

Skenario pada tabel 3.12 menggambarkan proses pencetakan dokumen resmi pengajuan SPBy atau SPM yang sudah mendapat persetujuan. Pengguna memilih pengajuan, melihat preview dokumen, lalu mencetak atau menyimpannya dalam format PDF.

Tabel 3. 12 Skenario *Use case* Mencetak Dokumen Pengajuan yang Disetujui

Skenario <i>Use case</i> 10: Mencetak Dokumen Pengajuan yang Disetujui	
Nomor	UC-10
Nama	Mencetak Dokumen Pengajuan yang Disetujui
Deskripsi	Proses mencetak dokumen resmi pengajuan SPBy atau SPM yang telah mendapat persetujuan.
<i>Primary Actor</i>	Bendahara Pengeluaran
<i>Preconditions</i>	Pengguna sudah <i>login</i> . Terdapat pengajuan dengan status "Disetujui".
<i>Postconditions</i>	Dokumen pengajuan berhasil dicetak atau disimpan dalam format PDF.
<i>Main success scenario</i>	1. Pengguna memilih menu Cetak Dokumen. 2. Sistem menampilkan daftar pengajuan yang telah disetujui. 3. Pengguna memilih pengajuan dan melihat preview dokumen 4. Pengguna mengklik tombol Cetak. 5. Sistem menghasilkan dokumen dan memulai proses pencetakan.
<i>Alternative Flow (Extension)</i>	4a. Pengguna dapat memilih simpan sebagai PDF jika tidak ada printer.

k. Skenario *Use case* Mengelola Master Data Anggaran

Skenario pada tabel 3.13 menjelaskan proses KPA dalam mengelola data master anggaran seperti program, kegiatan, sub kegiatan, dan akun belanja. KPA dapat menambah, mengubah, atau menghapus data, dan sistem akan memvalidasi sebelum perubahan disimpan.

Tabel 3. 13 Skenario *Use case* Mengelola Master Data Anggaran

Skenario <i>Use case</i> 11: Mengelola Master Data Anggaran	
Nomor	UC-11
Nama	Mengelola Master Data Anggaran
Deskripsi	Proses Kuasa Pengguna Anggaran mengelola data master anggaran meliputi program, kegiatan, sub kegiatan, dan akun belanja..
<i>Primary Actor</i>	Kuasa Pengguna Anggaran (KPA)
<i>Preconditions</i>	Kuasa Pengguna Anggaran sudah <i>login</i> . Tahun anggaran

	aktif sudah dikonfigurasi
<i>Postconditions</i>	Data master anggaran berhasil ditambah, diubah, atau dihapus.
<i>Main success scenario</i>	1. Pengguna memilih menu Master Data Anggaran. 2. Sistem menampilkan daftar data anggaran. 3. Pengguna memilih aksi (Tambah/Ubah/Hapus) dan mengisi data yang diperlukan 4. Pengguna mengklik tombol Simpan. 5. Sistem memvalidasi dan menyimpan perubahan.
<i>Alternative Flow (Extension)</i>	3a. Untuk Hapus, sistem menampilkan konfirmasi sebelum menghapus data. 5a. Jika kode anggaran sudah ada, sistem menampilkan pesan "Kode anggaran sudah digunakan".

1. Skenario *Use case* Melihat Semua Data Anggaran dan Realisasi

Skenario pada tabel 3.14 menjelaskan proses KPA dalam memantau seluruh data anggaran dan realisasi pembayaran secara menyeluruh. KPA dapat memfilter data berdasarkan tahun.

Tabel 3. 14 Skenario *Use case* Melihat Semua Data Anggaran dan Realisasi

Skenario <i>Use case</i> 12: Melihat Semua Data Anggaran dan Realisasi	
Nomor	UC-12
Nama	Melihat Semua Data Anggaran dan Realisasi
Deskripsi	Proses melihat seluruh data anggaran, penggunaan, dan realisasi pembayaran secara menyeluruh.
<i>Primary Actor</i>	Kuasa Pengguna Anggaran (KPA)
<i>Preconditions</i>	Kuasa Pengguna Anggaran sudah <i>login</i> . Data anggaran dan realisasi tersedia.
<i>Postconditions</i>	Pengguna dapat melihat rekap data anggaran dan realisasi sesuai filter yang dipilih.
<i>Main success scenario</i>	1. Pengguna memilih menu Data Anggaran & Realisasi. 2. Sistem menampilkan <i>dashboard</i> ringkasan anggaran (total pagu, terpakai, sisa). 3. Pengguna menentukan filter (tahun, unit, program, atau akun). 4. Sistem menampilkan data dalam bentuk tabel dan grafik. 5. Pengguna mengekspor data jika diperlukan.
<i>Alternative Flow (Extension)</i>	3a. Jika tidak ada data sesuai filter, sistem menampilkan pesan "Data tidak ditemukan" 5a. Jika ekspor gagal, sistem menampilkan pesan error.

m. Skenario *Use case* Mengakses Laporan Kegiatan Anggaran

Skenario pada tabel 3.15 menggambarkan proses pengguna dalam mengakses laporan kegiatan anggaran yang telah berjalan. Sistem menampilkan laporan secara terstruktur dan pengguna dapat mengeksponnya sesuai kebutuhan pelaporan institusi.

Tabel 3. 15 Skenario *Use case* Mengakses Laporan Kegiatan Anggaran

Skenario <i>Use case</i> 13: Mengakses Laporan Kegiatan Anggaran	
Nomor	UC-13
Nama	Mengakses Laporan Kegiatan Anggaran
Deskripsi	Proses mengakses dan mencetak laporan kegiatan anggaran dalam berbagai format pelaporan.
<i>Primary Actor</i>	Kuasa Pengguna Anggaran (KPA)
<i>Preconditions</i>	Kuasa Pengguna Anggaran sudah <i>login</i> . Data transaksi anggaran tersedia.
<i>Postconditions</i>	Laporan kegiatan anggaran berhasil ditampilkan dan dapat diunduh.
<i>Main success scenario</i>	1. Pengguna memilih menu Laporan Kegiatan Anggaran. 2. Sistem menampilkan pilihan jenis laporan (Realisasi, Progres Fisik, dll) 3. Pengguna memilih jenis laporan dan mengatur parameter 4. Sistem memproses dan menampilkan laporan. 5. Pengguna mengunduh atau mencetak laporan.
<i>Alternative Flow (Extension)</i>	5a. Pengguna dapat memilih format unduhan (PDF, Excel).

n. Skenario *Use case* Mengelola Mapping Pagu Unit dan PPK ke Akun

Skenario pada tabel 3.16 menjelaskan proses KPA dalam mengatur pemetaan antara pagu anggaran unit, PPK, dan akun belanja. KPA dapat menambah, mengubah, atau menghapus mapping, dan sistem akan memperingatkan jika total alokasi melebihi pagu akun yang tersedia.

Tabel 3. 16 Skenario *Use case* Mengelola Mapping Pagu Unit dan PPK ke Akun

Skenario <i>Use case</i> 14: Mengelola Mapping Pagu Unit dan PPK ke Akun	
Nomor	UC-14
Nama	Mengelola Mapping Pagu Unit dan PPK ke Akun
Deskripsi	Proses mengatur pemetaan antara pagu anggaran unit, PPK, dan akun belanja.
<i>Primary Actor</i>	Kuasa Pengguna Anggaran (KPA)
<i>Preconditions</i>	Kuasa Pengguna Anggaran sudah <i>login</i> . Data unit, PPK, dan akun anggaran tersedia.
<i>Postconditions</i>	Mapping pagu unit dan PPK ke akun berhasil

	dikonfigurasi.
<i>Main success scenario</i>	1. Pengguna memilih menu Mapping Pagu Unit & PPK. 2. Sistem menampilkan daftar mapping yang ada 3. Pengguna memilih aksi (Tambah/Ubah/Hapus) dan mengisi data mapping (unit, PPK, akun, nilai pagu) 4. Pengguna mengklik tombol Simpan. 5. Sistem memvalidasi dan menyimpan konfigurasi mapping.
<i>Alternative Flow (Extension)</i>	3a. Untuk Hapus, sistem menampilkan konfirmasi sebelum menghapus mapping. 5a. Jika total alokasi melebihi total pagu akun, sistem menampilkan peringatan.

o. Skenario *Use case* Mengelola Tahun Anggaran

Skenario pada tabel 3.17 menggambarkan proses KPA dalam mengatur konfigurasi tahun anggaran, termasuk membuka dan menutup periode anggaran.

Tabel 3. 17 Skenario *Use case* Mengelola Tahun Anggaran

Skenario <i>Use case</i> 15: Mengelola Tahun Anggaran	
Nomor	UC-15
Nama	Mengelola Tahun Anggaran
Deskripsi	Proses mengatur konfigurasi tahun anggaran aktif, membuka, dan menutup periode anggaran.
<i>Primary Actor</i>	Kuasa Pengguna Anggaran (KPA)
<i>Preconditions</i>	Kuasa Pengguna Anggaran sudah <i>login</i> .
<i>Postconditions</i>	Konfigurasi tahun anggaran berhasil diperbarui.
<i>Main success scenario</i>	1. Pengguna memilih menu Kelola Tahun Anggaran. 2. Sistem menampilkan daftar tahun anggaran yang tersedia. 3. Pengguna memilih aksi (Tambah/Ubah Status) dan mengisi data tahun anggaran 4. Pengguna mengklik tombol Simpan. 5. Sistem menyimpan dan memperbarui tahun anggaran aktif..
<i>Alternative Flow (Extension)</i>	3a. Untuk menutup tahun anggaran, sistem menampilkan konfirmasi bahwa aksi tidak dapat dibatalkan. 5a. Jika tahun anggaran sudah ada, sistem menampilkan pesan "Tahun anggaran sudah terdaftar".

p. Skenario *Use case* Mengelola Unit dan User

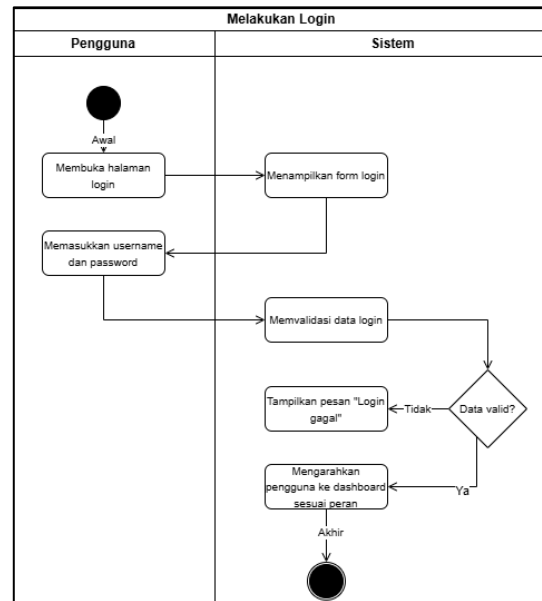
Skenario pada tabel 3.18 menjelaskan proses KPA dalam mengelola data unit organisasi dan akun pengguna sistem. KPA dapat menambah, mengubah, atau menonaktifkan user, dan sistem akan memblokir akses *login* bagi akun yang dinonaktifkan.

Tabel 3. 18 Skenario *Use case* Mengelola Unit dan User

Skenario <i>Use case</i> 16: Mengelola Unit dan User	
Nomor	UC-16
Nama	Mengelola Unit dan User
Deskripsi	Proses mengelola data unit organisasi dan akun pengguna sistem termasuk pembuatan akun, pengaturan peran, dan pengelolaan unit.
<i>Primary Actor</i>	Kuasa Pengguna Anggaran (KPA)
<i>Preconditions</i>	Kuasa Pengguna Anggaran sudah <i>login</i> .
<i>Postconditions</i>	Data unit atau akun pengguna berhasil ditambah, diubah, atau dinonaktifkan.
<i>Main success scenario</i>	1. Pengguna memilih menu Kelola Unit & User. 2. Sistem menampilkan tab Unit dan tab User. 3. Pengguna memilih tab yang diinginkan dan memilih aksi (Tambah/Ubah/Nonaktifkan) 4. Pengguna mengisi atau memperbarui data yang diperlukan. 5. Pengguna mengklik tombol Simpan. 6. Sistem menyimpan data dan menampilkan konfirmasi.
<i>Alternative Flow (Extension)</i>	4a. Jika <i>username</i> sudah digunakan, sistem menampilkan pesan "<i>Username</i> sudah terdaftar". 4b. Jika <i>email</i> sudah terdaftar, sistem menampilkan pesan "<i>Email</i> sudah digunakan". 3a. Untuk menonaktifkan user, sistem menampilkan konfirmasi dan memblokir akses <i>login</i>.

3.2.6 Activity diagram

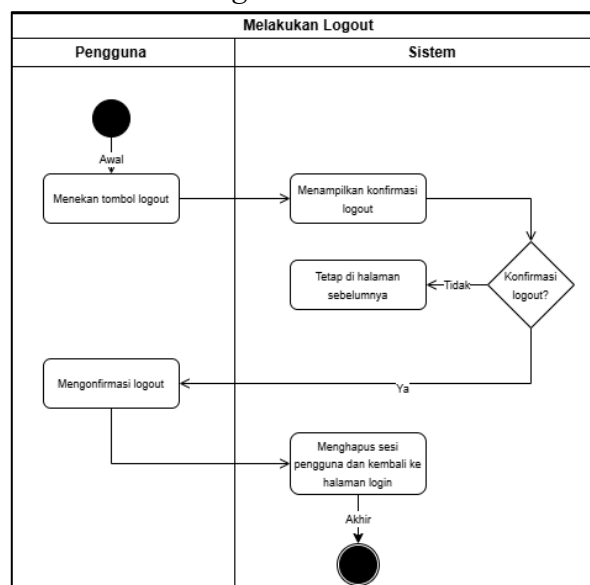
a. Activity diagram Melakukan Login



Gambar 3. 4 Activity diagram Melakukan Login

Activity diagram ini menggambarkan proses autentikasi pengguna ke dalam sistem. Alur utama adalah pengujian validitas kredensial di mana kegagalan validasi akan mengarahkan pengguna untuk mengulang input.

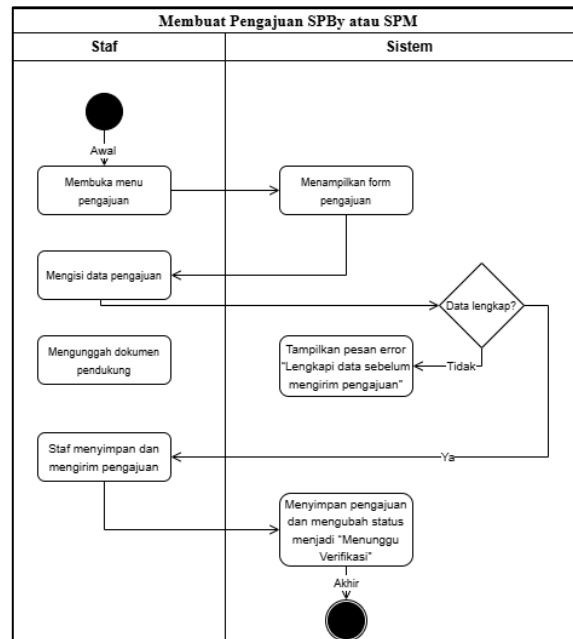
b. Activity diagram Melakukan Logout



Gambar 3. 5 Activity diagram Melakukan Logout

Diagram ini menunjukkan langkah-langkah pengguna untuk mengakhiri sesi penggunaan sistem secara aman, termasuk opsi pembatalan *logout*.

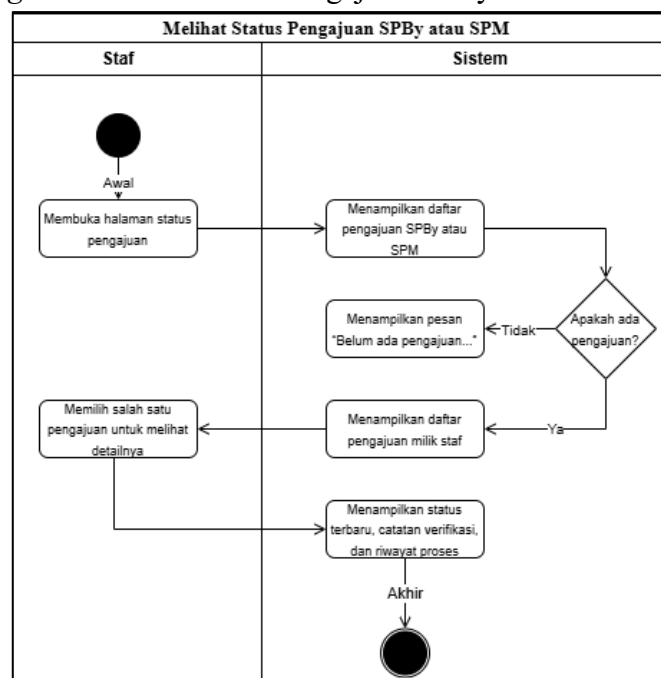
c. *Activity diagram* Membuat Pengajuan SPBy atau SPM LS



Gambar 3. 6 *Activity diagram* Membuat Pengajuan SPBy atau SPM

Diagram ini fokus pada alur kerja Staf dalam membuat pengajuan baru. Keputusan utama dalam alur ini adalah validasi kelengkapan data dan dokumen pendukung sebelum pengajuan dikirimkan untuk verifikasi.

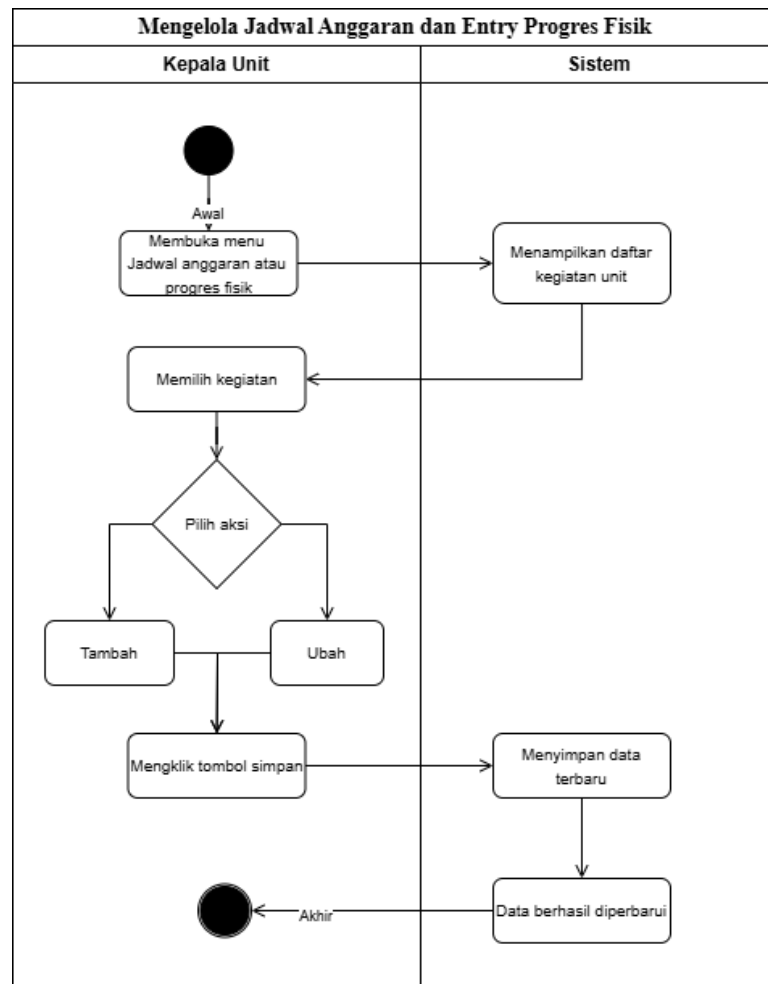
d. *Activity diagram* Melihat Status Pengajuan SPBy atau SPM LS



Gambar 3. 7 *Activity diagram* Melihat Status Pengajuan SPBy atau SPM

Diagram ini menggambarkan alur kerja Staf dalam memantau perkembangan pengajuan yang telah mereka buat, termasuk penanganan jika belum ada pengajuan sama sekali.

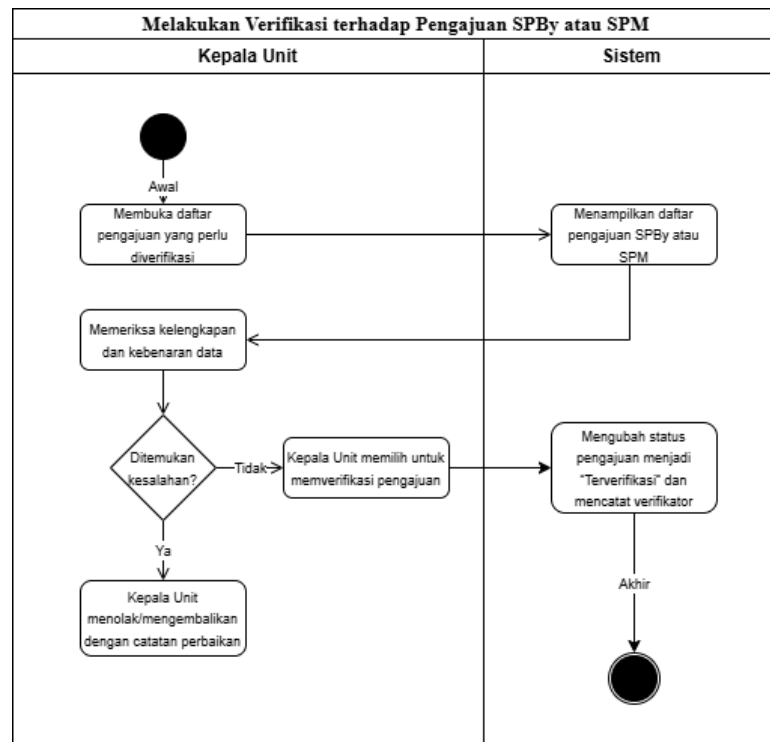
e. *Activity diagram* Mengelola Jadwal Anggaran dan Entry Progres Fisik



Gambar 3. 8 *Activity diagram* Mengelola Jadwal Anggaran dan Entry Progres Fisik

Diagram ini menggambarkan alur kerja Kepala Unit dalam mengatur jadwal pelaksanaan anggaran dan mencatat progres fisik kegiatan yang sedang berjalan. Keputusan utama dalam alur ini adalah validasi nilai progres fisik dan rentang tanggal, di mana sistem akan menampilkan peringatan jika data yang dimasukkan tidak sesuai sebelum dapat disimpan.

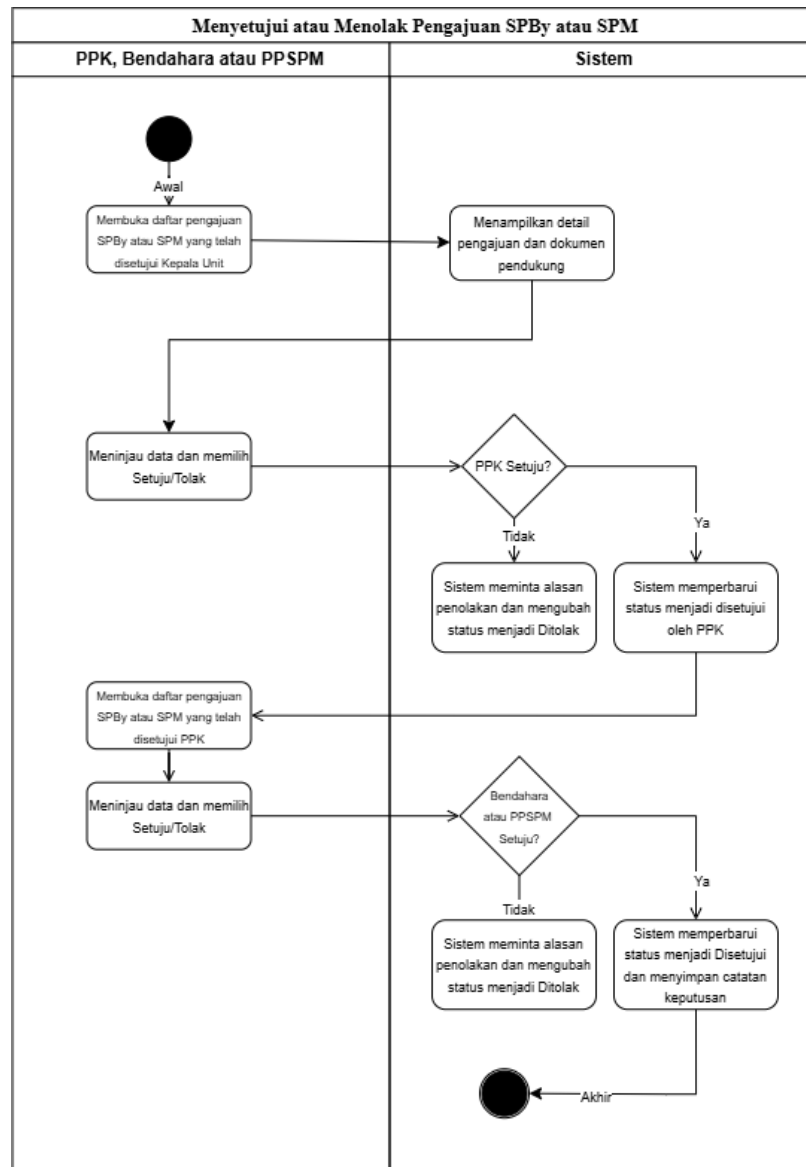
- f. *Activity diagram* Melakukan Verifikasi terhadap Pengajuan SPBy atau SPM LS



Gambar 3. 9 *Activity diagram* Melakukan Verifikasi terhadap Pengajuan SPBy atau SPM

Diagram ini berfokus pada peran Kepala Unit dalam melakukan verifikasi awal. Alur utama melibatkan pemeriksaan kelengkapan data dengan opsi untuk memverifikasi atau mengembalikan (menolak) pengajuan.

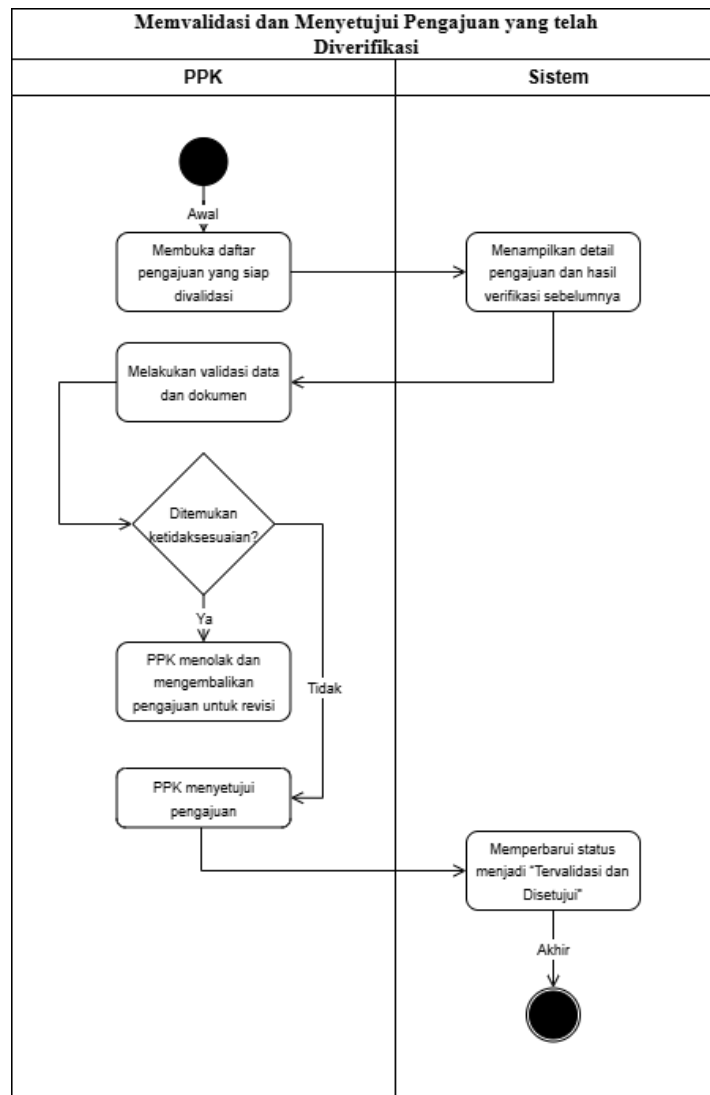
g. *Activity diagram* Menyetujui atau Menolak Pengajuan SPBy atau SPM



Gambar 3. 10 *Activity diagram* Menyetujui atau Menolak Pengajuan SPBy atau SPM

Diagram ini menunjukkan proses persetujuan bertingkat yang melibatkan Kepala Unit, PPK, Bendahara atau PPSPM secara berurutan. Setiap aktor memiliki hak untuk menolak pengajuan, yang akan mengakhiri alur persetujuan.

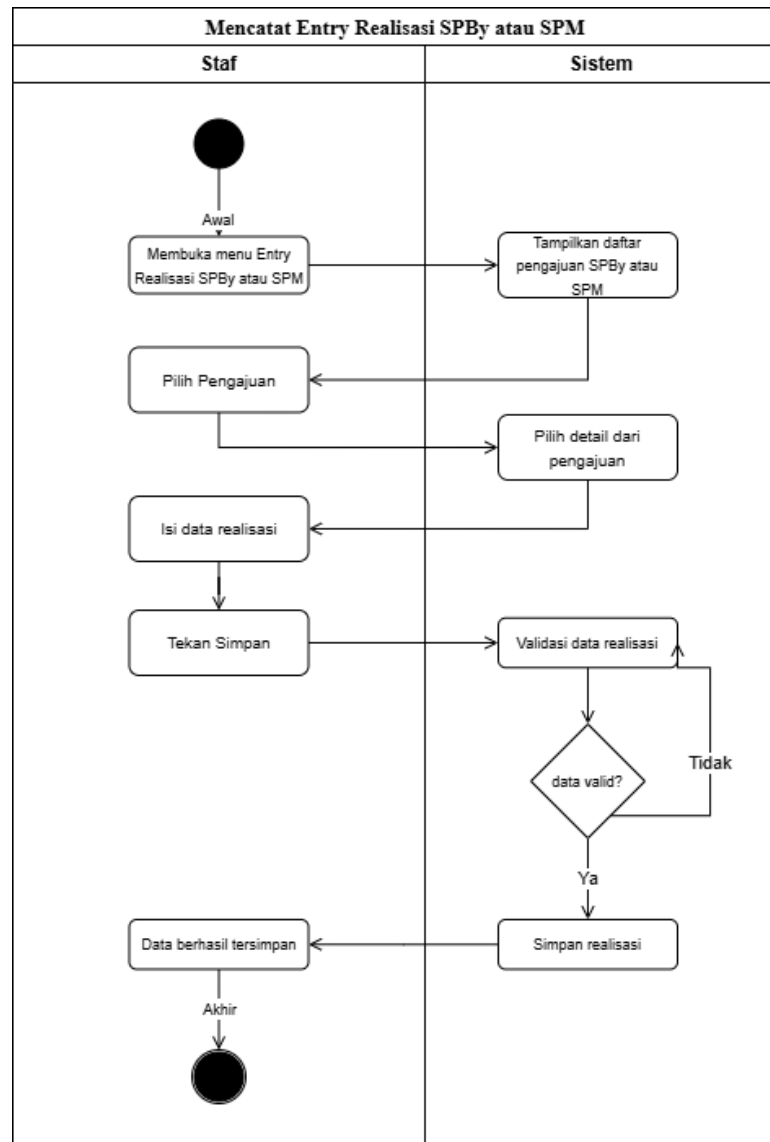
- h. *Activity diagram* Memvalidasi dan Menyetujui Pengajuan yang telah Diverifikasi



Gambar 3. 11 *Activity diagram* Memvalidasi dan Menyetujui Pengajuan yang telah Diverifikasi

Diagram ini menguraikan peran PPK dalam memberikan validasi akhir terhadap pengajuan (SPBy atau SPM) yang sudah diverifikasi, memastikan semua persyaratan telah terpenuhi sebelum status persetujuan penuh diberikan.

i. *Activity diagram* Mencatat Entry Realisasi SPBy atau SPM



Gambar 3. 12 *Activity diagram* Mencatat Entry Realisasi SPBy atau SPM

Diagram ini menjelaskan proses Staf dalam mencatat realisasi penggunaan anggaran berdasarkan pengajuan SPBy GU atau SPM LS yang telah disetujui, mulai dari memilih pengajuan, mengisi data realisasi, hingga data tersimpan dan sisa anggaran diperbarui pada sistem.

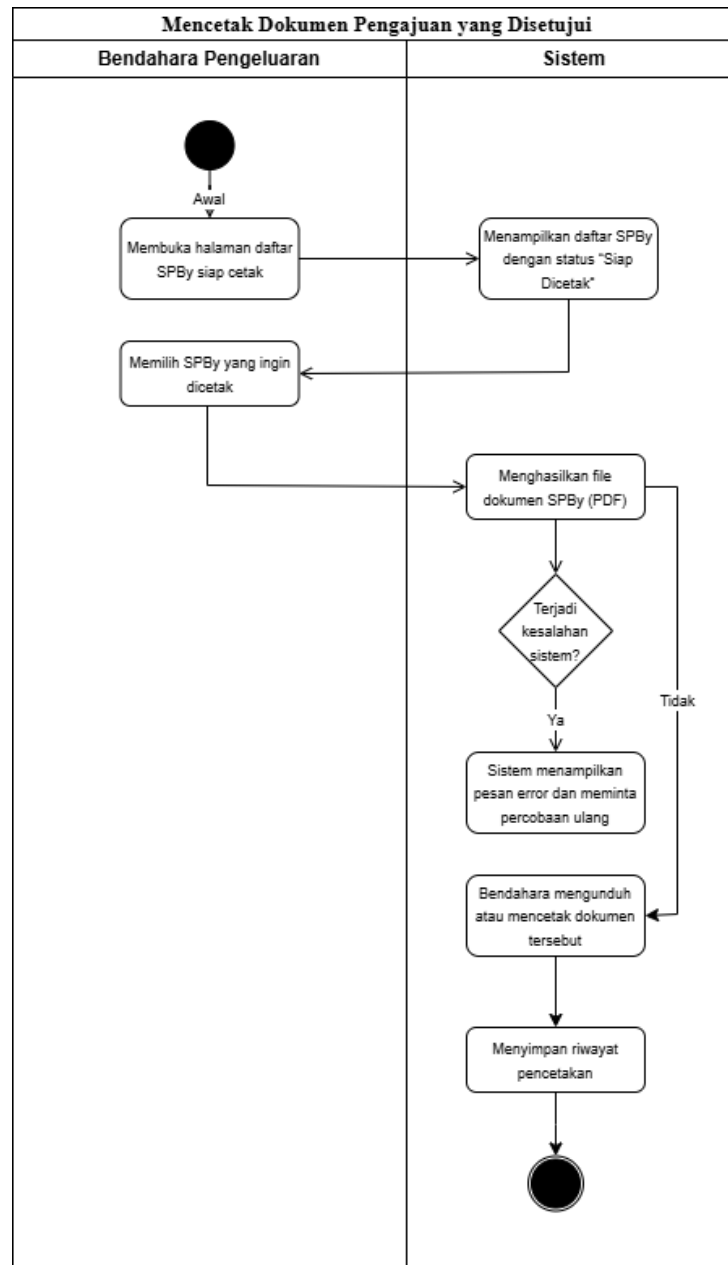
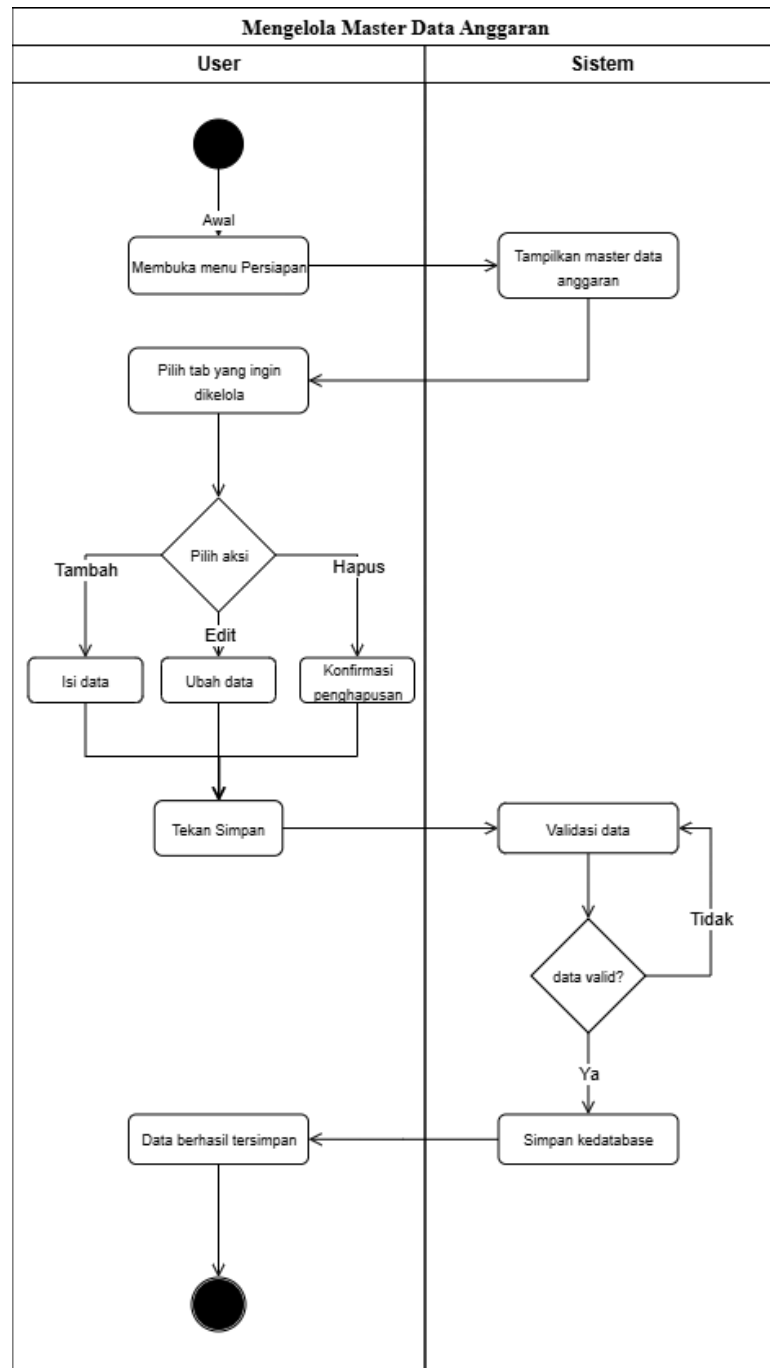
j. *Activity diagram* Mencetak Dokumen Pengajuan yang DisetujuiGambar 3. 13 *Activity diagram* Mencetak Dokumen Pengajuan yang Disetujui

Diagram ini menggambarkan langkah-langkah yang dilakukan Bendahara Pengeluaran untuk mencetak dokumen SPBy yang sudah "Siap Dicetak", termasuk penanganan jika terjadi error saat pembuatan dokumen.

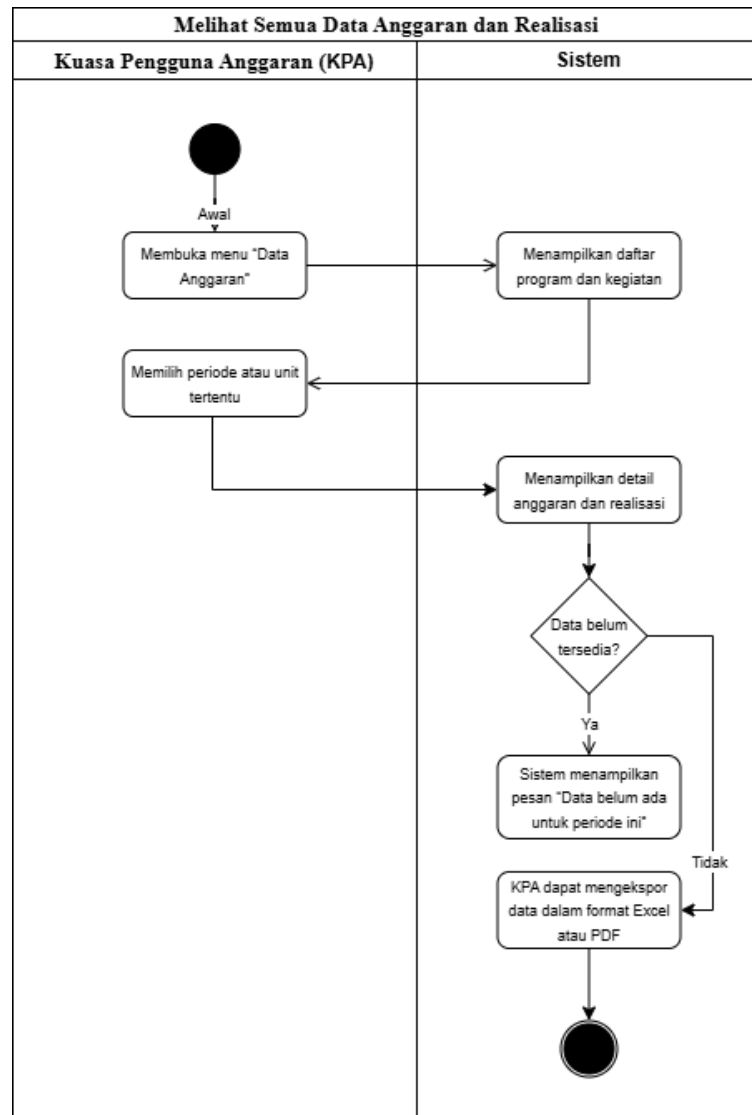
k. *Activity diagram* Mengelola Master Data Anggaran



Gambar 3. 14 *Activity diagram* Mengelola Master Data Anggaran

Diagram ini menjelaskan User tertentu dalam mengelola data hierarki anggaran, mulai dari menambah, mengubah, hingga menghapus data pada menu Master Data Anggaran.

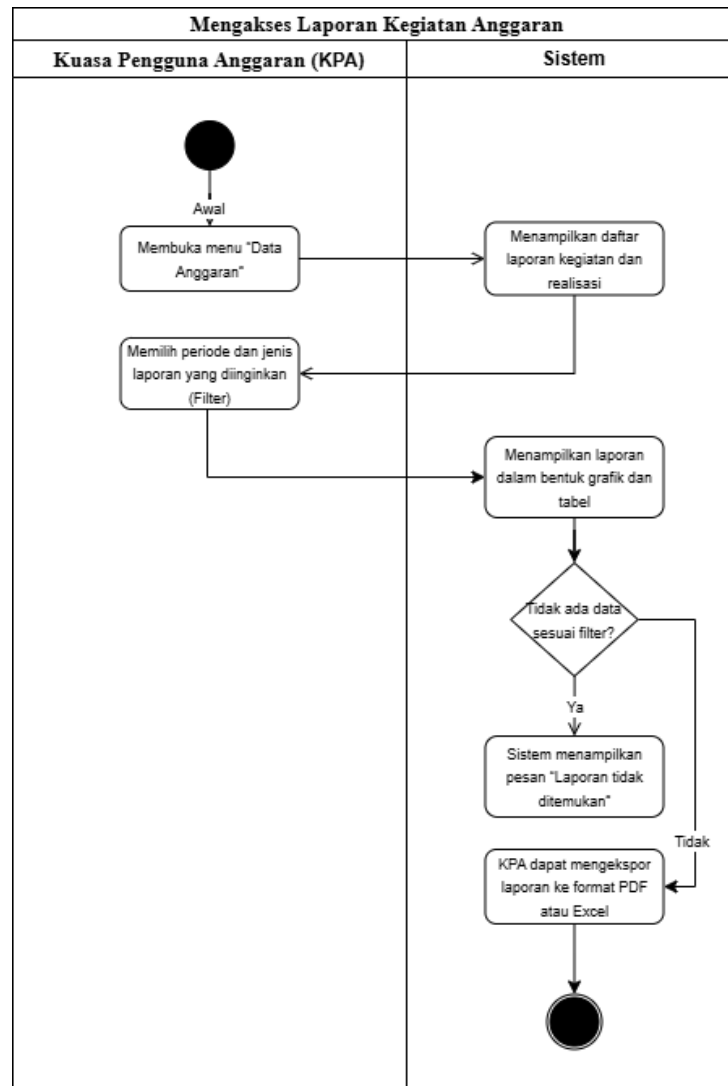
1. *Activity diagram* Melihat Semua Data Anggaran dan Realisasi



Gambar 3. 15 *Activity diagram* Melihat Semua Data Anggaran dan Realisasi

Diagram ini menunjukkan alur kerja Kuasa Pengguna Anggaran (KPA) dalam mengakses data anggaran, menggunakan filter periode/unit, dan opsi untuk mengekspor data.

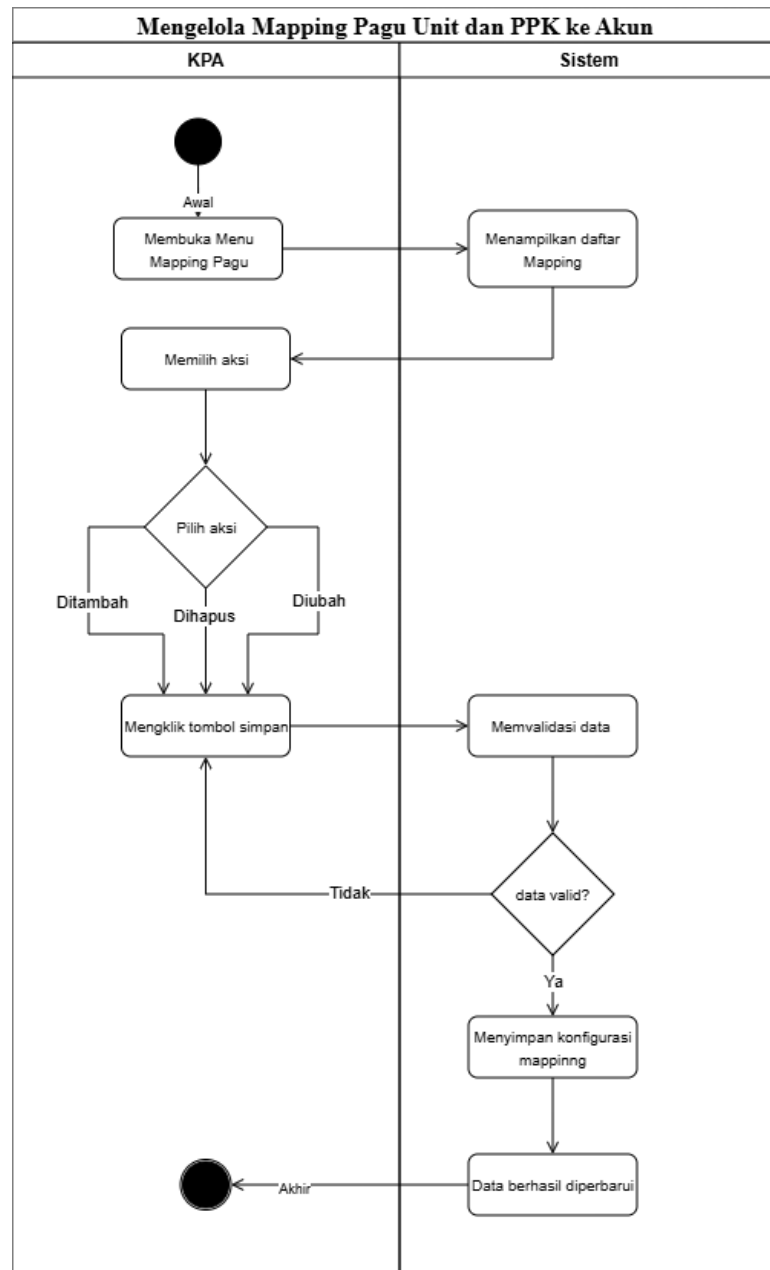
m. *Activity diagram* Mengakses Laporan Kegiatan Anggaran



Gambar 3. 16 *Activity diagram* Mengakses Laporan Kegiatan Anggaran

Diagram ini menguraikan langkah-langkah KPA dalam menghasilkan laporan anggaran berdasarkan kriteria filter, serta penanganan jika laporan yang dicari tidak ditemukan.

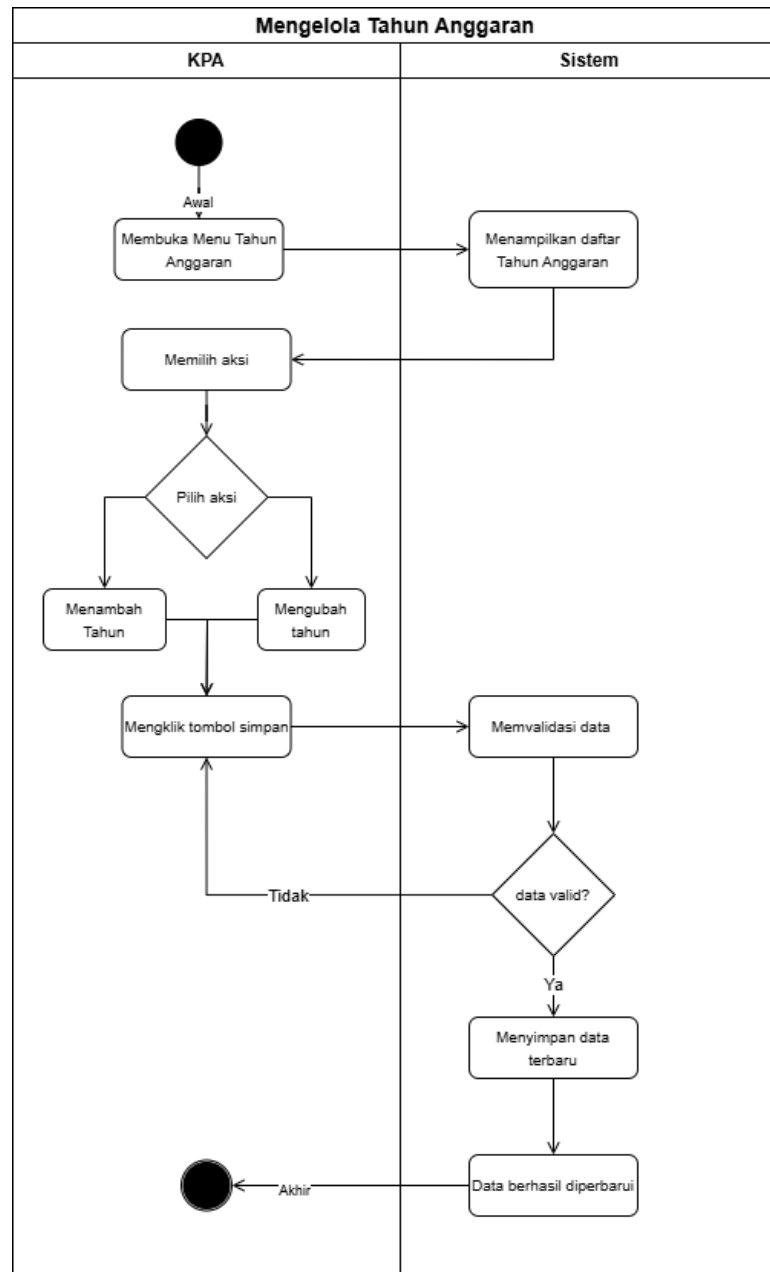
n. *Activity diagram* Mengelola Mapping Pagu Unit dan PPK ke Akun



Gambar 3. 17 *Activity diagram* Mengelola Mapping Pagu Unit dan PPK ke Akun

Diagram ini menggambarkan alur KPA dalam memetakan pagu anggaran unit dan PPK ke akun belanja. Sistem memvalidasi total alokasi sebelum menyimpan dan menampilkan peringatan jika melebihi batas pagu.

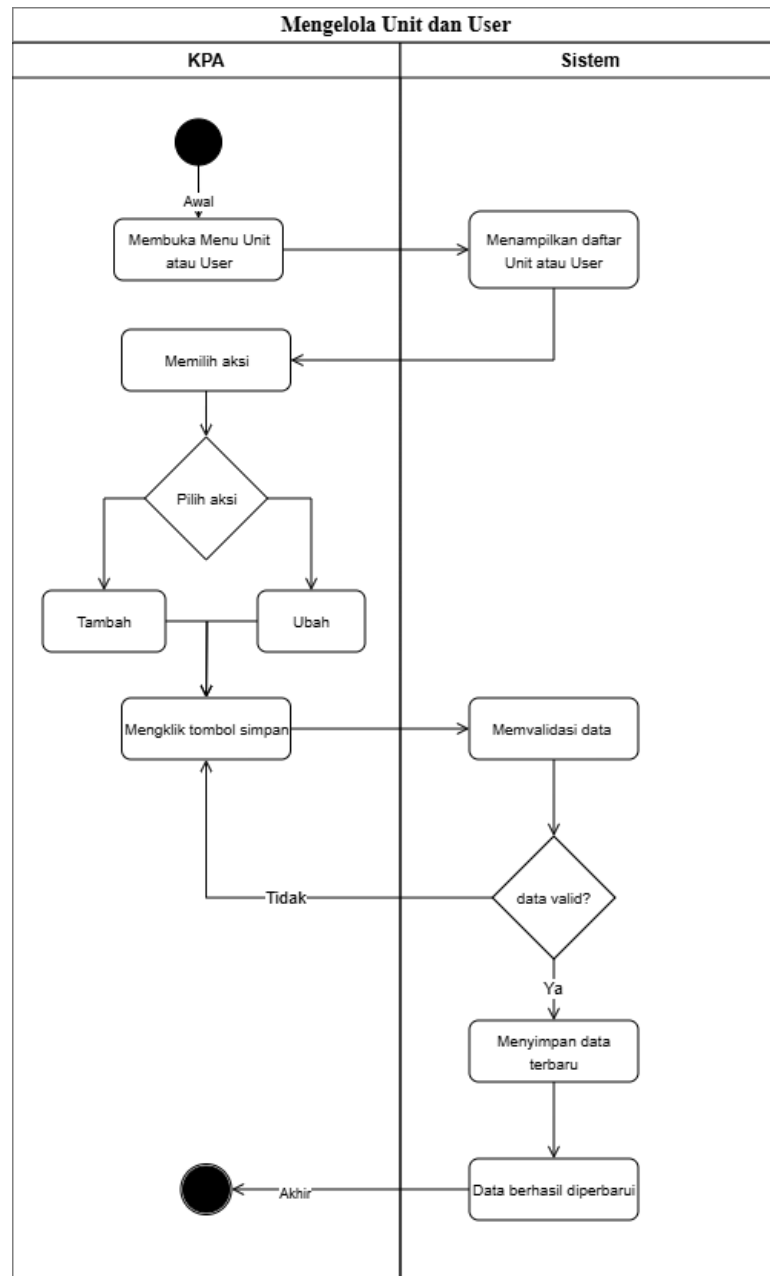
o. *Activity diagram* Mengelola Tahun Anggaran



Gambar 3. 18 *Activity diagram* Mengelola Tahun Anggaran

Diagram ini menggambarkan alur KPA dalam mengelola konfigurasi periode tahun anggaran. Keputusan utama adalah perubahan status tahun anggaran, di mana sistem meminta konfirmasi sebelum menutup periode yang tidak dapat dibatalkan.

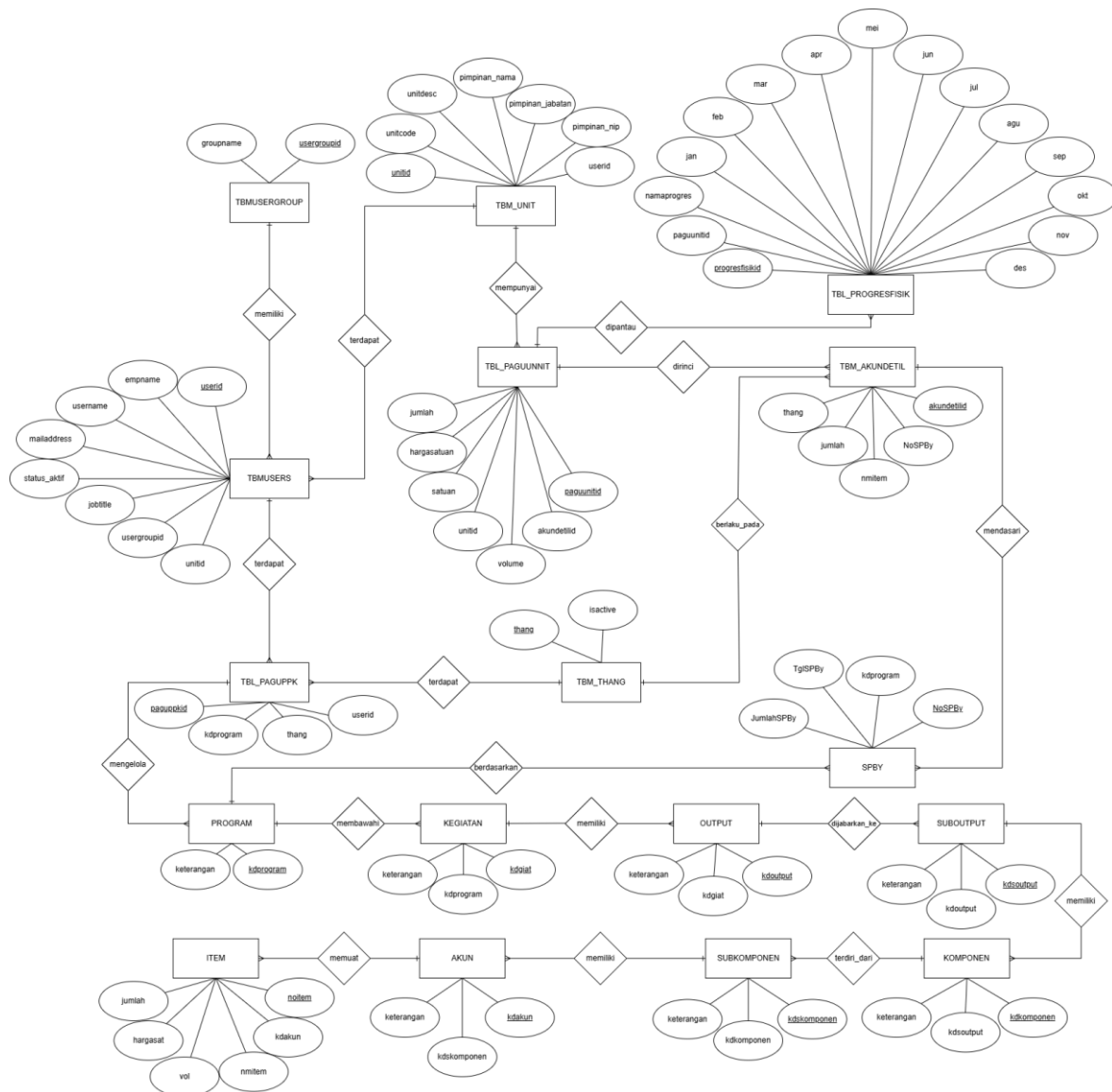
p. *Activity diagram* Mengelola Unit dan User



Gambar 3. 19 *Activity diagram* Mengelola Unit dan User

Diagram ini menggambarkan alur KPA dalam menambah, mengubah, atau menonaktifkan data unit dan akun pengguna. Sistem memvalidasi keunikan *username* dan *email*, serta memblokir akses *login* bagi user yang dinonaktifkan.

3.2.7 Entity relationship diagram

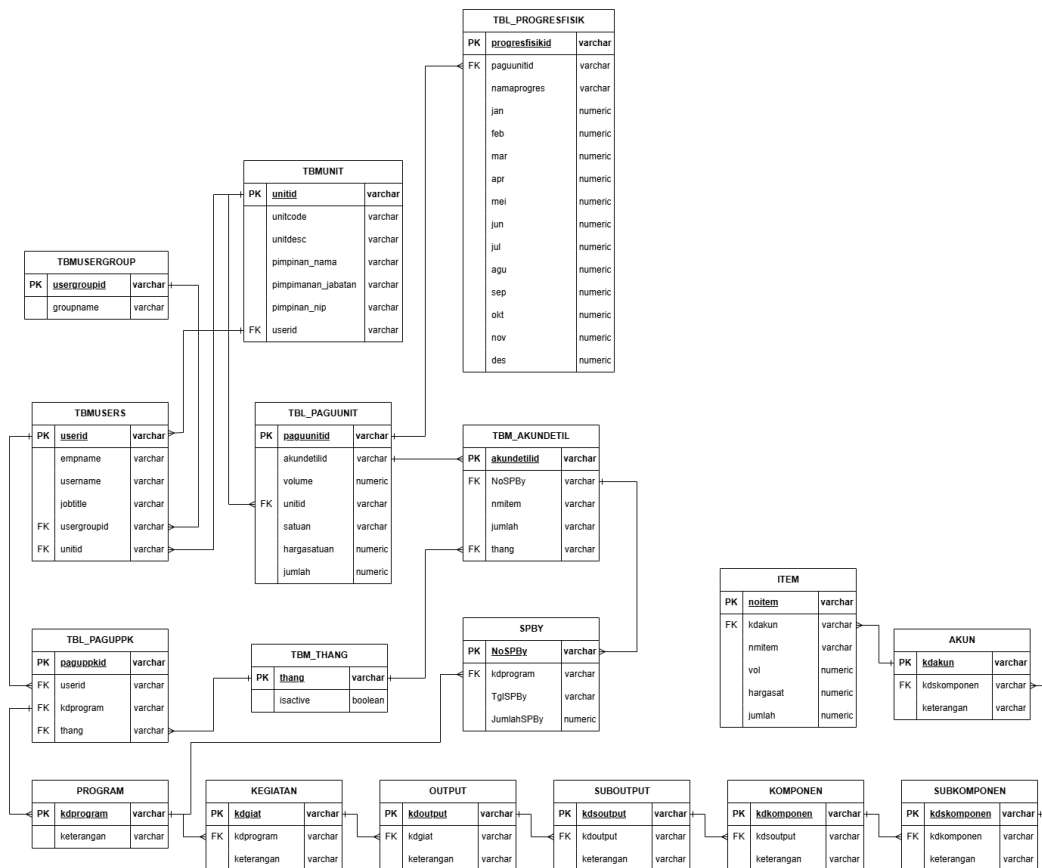


Gambar 3. 20 Entity relationship diagram SIPA

Gambar 3.20 di atas merupakan *Entity relationship diagram* (ERD) yang memodelkan struktur basis data relasional pada Sistem Informasi Pengelolaan Anggaran (SIPA). Sistem ini menempatkan entitas TBM_UNIT sebagai fondasi utama yang berelasi dengan TBM_USERS untuk mencatat pengguna sistem dan TBMUSERGROUP untuk mengatur hak akses, sementara distribusi anggaran dikelola melalui TBL_PAGUUNIT yang terhubung dengan TBM_AKUNDETIL sebagai rincian pos akun belanja dan TBL_PROGRESFISIK untuk memantau realisasi fisik bulanan. Hierarki anggaran dibangun melalui rantai relasi berjenjang mengikuti struktur RKAKL yakni Program, Kegiatan, Output, Suboutput, Komponen, Subkomponen, Akun, Item, sehingga setiap belanja dapat

ditelusuri hingga ke rincian paling dasar, dengan pertanggungjawaban keuangan yang diakomodasi melalui entitas SPBY serta kewenangan PPK yang dikelola melalui TBL_PAGUPPK dengan TBM_THANG sebagai referensi tahun anggaran yang mengikat seluruh transaksi dalam sistem.

3.2.8 Relational Schema



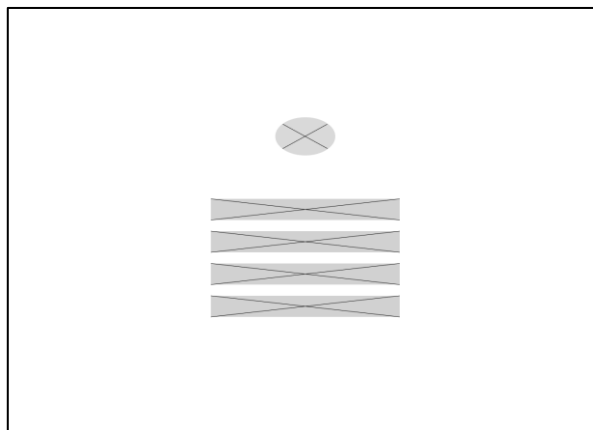
Gambar 3. 21 Relational Schema

Gambar 3.21 di atas merupakan Skema Relasional yang merepresentasikan struktur basis data pada Sistem Informasi Pengelolaan Anggaran (SIPA) dalam bentuk tabel-tabel beserta atribut dan keterhubungannya secara eksplisit. Skema ini menempatkan entitas TBM_UNIT sebagai referensi pusat data satuan kerja yang menyimpan informasi pimpinan unit, di mana entitas ini berelasi dengan TBMUSERS untuk mengelola data pengguna sistem, sementara hak akses setiap pengguna diatur melalui entitas TBMUSERGROUP menggunakan atribut usergroupid sebagai *foreign key*. Alokasi anggaran per unit dikelola melalui entitas TBL_PAGUUNIT yang menghubungkan unitid dari TBMUNIT dengan akundetildid dari TBM_AKUNDETIL, sehingga setiap pos anggaran dapat dilacak

realisasi fisiknya secara bulanan melalui entitas TBL_PROGRESFISIK menggunakan paguunitid sebagai kunci penghubung. Hierarki perencanaan anggaran direpresentasikan melalui rantai relasi bertingkat yang membentuk struktur RKAKL, yaitu Program, Kegiatan, Output, Suboutput, Komponen, Subkomponen, Akun, Item, di mana setiap level dihubungkan melalui kode referensi sebagai *foreign key* sehingga penelusuran anggaran dari tingkat program hingga rincian item belanja dapat dilakukan secara menyeluruh. Pertanggungjawaban keuangan dikelola melalui entitas SPBY yang berelasi dengan TBM_AKUNDETIL menggunakan NoSPBy sebagai penghubung, sedangkan entitas TBL_PAGUPPK mencatat kewenangan pengelolaan anggaran oleh Pejabat Pembuat Komitmen dengan mengacu pada userid, kdprogram, dan thang, di mana entitas TBM_THANG berfungsi sebagai referensi tahun anggaran yang mengikat seluruh transaksi keuangan dalam sistem. Secara keseluruhan, skema relasional ini menggambarkan integritas referensial yang kuat antar seluruh tabel sehingga menjamin konsistensi dan ketertelusuran data dalam siklus pengelolaan anggaran secara *end-to-end*.

3.2.9 Perancangan Antarmuka (*Wireframe*)

a. *Wireframe Login*

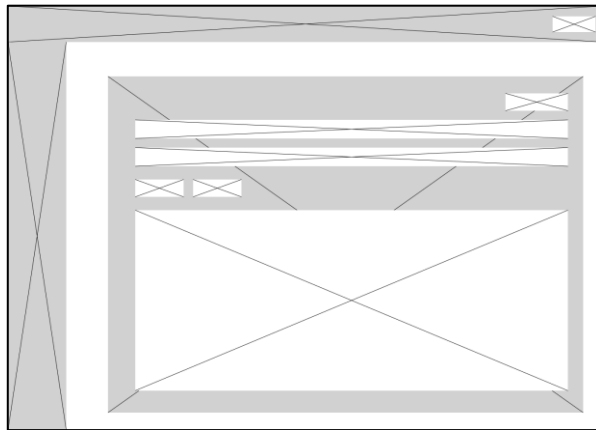


Gambar 3. 22 *Wireframe Login*

Wireframe login merupakan halaman awal yang digunakan oleh pengguna untuk masuk ke dalam sistem. Pada halaman ini, pengguna diminta untuk memasukkan nama pengguna (*username*), kata sandi (*password*), serta tahun anggaran yang digunakan. Setelah seluruh data

diisi dengan benar, pengguna dapat menekan tombol “*Login*” untuk mengakses sistem.

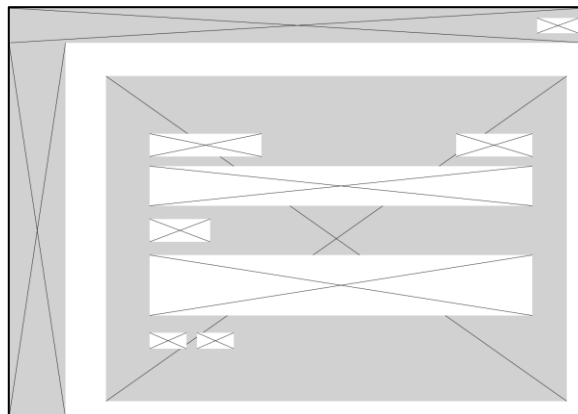
b. *Wireframe* Pengajuan SPBy atau SPM



Gambar 3. 23 *Wireframe* Pengajuan SPBy atau SPM

Wireframe ini menampilkan daftar seluruh pengajuan yang telah dibuat oleh staff. Setiap data pengajuan menampilkan informasi seperti nomor pengajuan, tanggal pengajuan, nama kegiatan, serta status pengajuan.

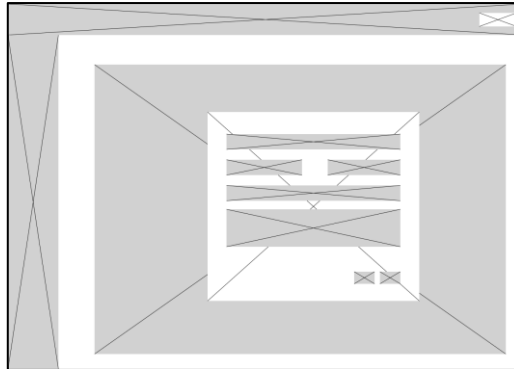
c. *Wireframe* Formulir Pengajuan Baru



Gambar 3. 24 *Wireframe* Formulir Pengajuan Baru

Wireframe ini merupakan halaman setelah pengguna menekan tombol “Tambah” pada Pengajuan SPBy atau SPM, sistem akan menampilkan halaman formulir pengajuan baru. Pada halaman ini, pengguna diminta untuk mengisi informasi utama pengajuan,

d. *Wireframe* Tambah Item

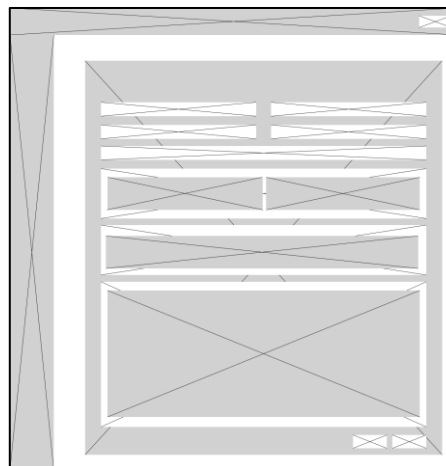


Gambar 3. 25 *Wireframe* Tambah Item

Pada *Wireframe* ini merupakan halaman setelah pengguna menekan tombol “tambah item” pada *Wireframe* Formulir Pengajuan Baru yang berfungsi untuk menambahkan rincian detail dari pengajuan. Setelah tombol ini diklik, sistem akan menampilkan *pop-up* form yang berisi kolom-kolom isian seperti kegiatan, kode akun, sisa anggaran, jumlah uang, dan keterangan.

Setelah pengguna mengisi seluruh data item, tombol “OK” ditekan untuk menyimpan item tersebut. Item yang sudah ditambahkan akan muncul dalam daftar di bawah form utama. Setelah seluruh data selesai diisi, pengguna dapat menekan tombol “Simpan dan Ajukan” untuk mengirim pengajuan.

e. *Wireframe* Entry Realisasi SPBy GU atau SPM LS

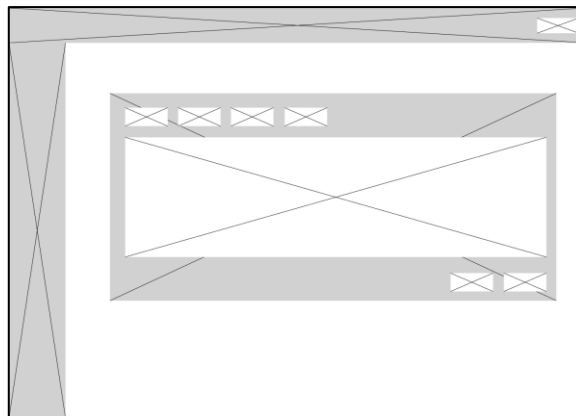


Gambar 3. 26 *Wireframe* Entry Realisasi SPBy GU atau SPM LS

Pada *Wireframe* ini ditampilkan halaman yang digunakan untuk melakukan proses verifikasi, persetujuan, atau penolakan terhadap

dokumen SPBy maupun SPM. Pada halaman ini, Kepala Unit, PPK, dan Bendahara dapat melihat informasi SPBy secara detail. Selain itu, apabila dokumen tersebut berupa SPM LS, maka PPSPM dapat meninjau dan memberikan persetujuan apabila data yang ditampilkan telah sesuai.

f. *Wireframe* Laporan Kegiatan Anggaran



Gambar 3. 27 *Wireframe* Laporan Kegiatan Anggaran

Halaman ini digunakan untuk menampilkan laporan realisasi anggaran secara detail berdasarkan program, kegiatan, output, dan komponen. Terdapat beberapa tab laporan seperti *Summary* Realisasi Anggaran, Realisasi Per Level, Realisasi Per Unit, dan Rekap Realisasi,

3.2.10 Jadwal Penelitian

Jadwal penelitian ini disusun berdasarkan kegiatan nyata selama proses magang dalam proyek migrasi Sistem Informasi Pengelolaan Anggaran dari bahasa pemrograman C# ke PHP, mulai dari analisis awal, migrasi antarmuka, implementasi backend, hingga penyusunan laporan akhir.

Tabel 3. 19 Jadwal Penelitian

No	Kegiatan	Periode
1	Observasi dan pengenalan sistem lama	8–15 September 2025
2	Pemahaman alur sistem dan analisis	15-22 September 2025
3	Migrasi tampilan aplikasi	22 September –5 Oktober 2025
4	Penataan database dan implementasi Backend	6-12 Oktober 2025
5	Penyusunan dokumen analisis dan perancangan	13-24 Oktober 2025
6	Implementasi keseluruhan migrasi dari aplikasi SIPA	27 Oktober 2025 – 28 Februari 2026

7	<i>Testing</i> dan <i>deployment</i> aplikasi SIPA	2 Maret – 30 April 2026
8	Penyusunan laporan akhir	4 mei -15 Juni 2026

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

Sistem Informasi Pengelolaan Anggaran di Politeknik Negeri Batam dibuat sebagai aplikasi berbasis web. Aplikasi ini menggunakan framework Laravel dan bahasa pemrograman PHP. Sistem ini dibuat untuk menggantikan sistem lama yang menggunakan bahasa C#. Tujuan utama dari pembuatan sistem ini adalah untuk menciptakan sistem yang stabil, mudah dipelihara, dan dapat dikelola sendiri oleh tim *internal* Politeknik Negeri Batam.

Sistem SIPA dapat diakses dengan dua cara. Pertama, melalui Web App yang digunakan oleh Staf dan Kepala Unit. Mereka menggunakan Web App untuk mengajukan anggaran, mencatat progres fisik, dan memasukkan realisasi. Kedua, melalui Desktop App yang digunakan oleh PPK, Bendahara Pengeluaran, PPSPM, dan KPA. Mereka menggunakan Desktop App untuk mengelola master data, verifikasi, dan pelaporan anggaran. Kedua cara akses ini terintegrasi dalam satu sistem dan menggunakan basis data yang sama. Dengan demikian, semua data anggaran dapat dipantau secara *real-time* oleh pihak yang berwenang.

Berikut adalah tampilan antarmuka dari sistem SIPA yang telah diimplementasikan beserta penjelasan fungsi pada setiap halamannya.

a. Login



Gambar 4. 1 Tampilan *Login*

Halaman *Login* merupakan halaman awal yang ditampilkan ketika pengguna mengakses sistem SIPA. Pengguna diwajibkan memasukkan *username* dan *password* sesuai dengan peran yang telah didaftarkan untuk dapat masuk ke dalam sistem.

Gambar 4. 4 Tampilan Halaman Pengajuan SPM LS
Halaman Pengajuan SPBy GU atau SPM LS digunakan oleh Staf untuk mengajukan permintaan anggaran. Halaman ini menampilkan daftar pengajuan

yang telah dibuat beserta status prosesnya, sehingga Staf dapat memantau perkembangan setiap pengajuan yang sedang berjalan.

d. Menambah pengajuan SPBy atau SPM LS

Gambar 4. 5 Tampilan Form Pengajuan SPBy

Gambar 4. 6 Tampilan Form Pengajuan SPM LS

Halaman Menambah Pengajuan merupakan formulir yang digunakan oleh Staf untuk membuat pengajuan SPBy GU atau SPM LS baru. Staf mengisi data yang diperlukan seperti item belanja, jumlah, dan keterangan, kemudian mengirimkan pengajuan untuk diproses lebih lanjut oleh pihak yang berwenang.

e. Entry Realisasi SPBy atau SPM LS

No	Kode Regimen	Kode Akun	Jumlah Uang	Keterangan	Status	Tanggal
1	00001	00001	10000000	Belanja Pegawai	Selesai	2020-01-01
2	00002	00002	20000000	Belanja Pegawai	Selesai	2020-01-02
3	00003	00003	30000000	Belanja Pegawai	Selesai	2020-01-03
4	00004	00004	40000000	Belanja Pegawai	Selesai	2020-01-04
5	00005	00005	50000000	Belanja Pegawai	Selesai	2020-01-05
6	00006	00006	60000000	Belanja Pegawai	Selesai	2020-01-06
7	00007	00007	70000000	Belanja Pegawai	Selesai	2020-01-07
8	00008	00008	80000000	Belanja Pegawai	Selesai	2020-01-08
9	00009	00009	90000000	Belanja Pegawai	Selesai	2020-01-09
10	00010	00010	100000000	Belanja Pegawai	Selesai	2020-01-10

Gambar 4. 7 Tampilan Halaman Entry Realisasi SPBy

Halaman Entry Realisasi SPBy GU atau SPM LS digunakan oleh Staf

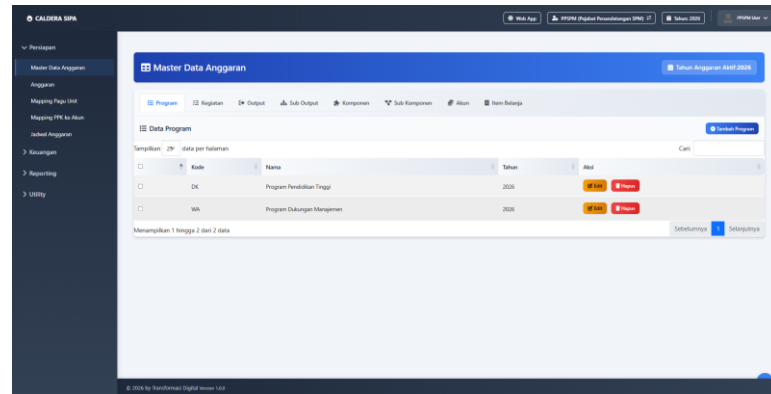
f. Verifikasi pengajuan SPBy atau SPM LS

[illegible]

Halaman Verifikasi Pengajuan digunakan oleh Kepala Unit atau PPK

untuk memeriksa dan memberikan keputusan terhadap pengajuan yang masuk. Pengguna dapat menyetujui atau menolak pengajuan disertai catatan apabila diperlukan.

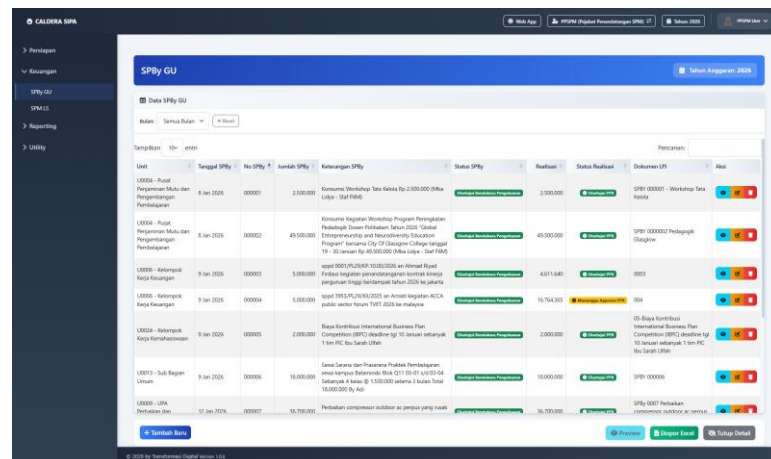
g. Tampilan Master data anggaran



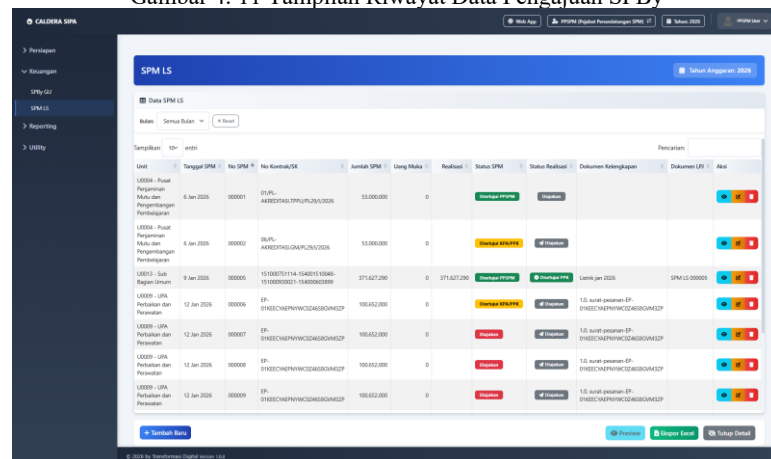
Gambar 4. 10 Tampilan Master Data Anggaran

Halaman Master Data Anggaran pada Desktop App digunakan oleh PPK atau PPSPM untuk mengelola data hierarki anggaran yang meliputi Program, Kegiatan, Output, Sub Output, Komponen, Sub Komponen, Akun, dan Item Belanja sebagai dasar penyusunan anggaran institusi.

h. Halaman Semua Riwayat Data SPBy atau SPM LS



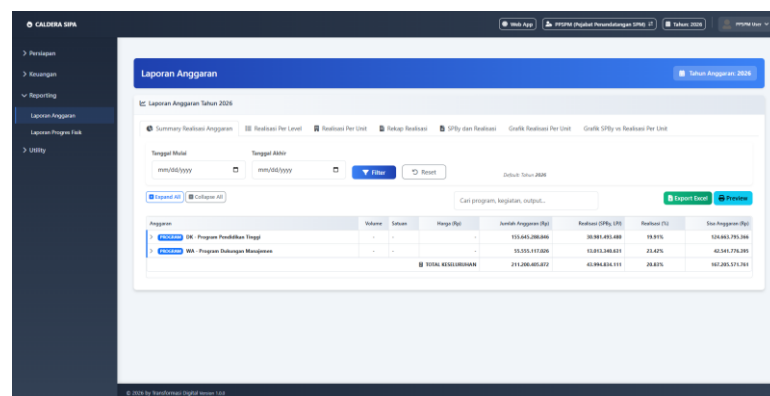
Gambar 4. 11 Tampilan Riwayat Data Pengajuan SPBy



Gambar 4. 12 Tampilan Riwayat Data Pengajuan SPM LS

Halaman ini menampilkan seluruh riwayat pengajuan SPBy GU dan SPM LS yang pernah diproses dalam sistem. Pengguna yang berwenang seperti Bendahara dan PPSPM dapat memantau status setiap pengajuan secara menyeluruh melalui halaman ini.

i. Laporan kegiatan anggaran



Gambar 4. 13 Tampilan Laporan Kegiatan Anggaran

Halaman Laporan Kegiatan Anggaran menyediakan rekam data seluruh kegiatan anggaran yang telah berjalan. Pengguna yang memiliki akses ke menu Reporting dapat melihat dan mengunduh laporan sebagai bahan evaluasi dan pertanggungjawaban penggunaan anggaran institusi.

4.2. Pengujian

Pengujian perangkat lunak pada sistem ini dilakukan menggunakan metode *Black Box Testing*. Metode ini bertujuan untuk mengevaluasi fungsionalitas aplikasi berdasarkan *input* dan *output* pada antarmuka, tanpa melihat struktur kode program di baliknya. Pengujian ini bertujuan untuk membuktikan bahwa seluruh Kebutuhan Fungsional yang telah dirancang pada Sistem Informasi Pengelolaan Anggaran (SIPA) dapat beroperasi secara normal, mampu menangani *error*, dan memberikan respons yang sesuai berdasarkan peran pengguna, baik Staf, Kepala Unit, PPK, Bendahara, PPSPM, maupun KPA. Berikut adalah rincian hasil pengujian fungsionalitas sistem menggunakan metode *Black Box Testing*.

a. Test case pengguna melakukan login

Pengujian *Black Box* pada fitur *login* bertujuan memastikan sistem bisa menjalankan proses autentikasi pengguna dengan baik. Pengujian ini meliputi

beberapa skenario: *login* menggunakan kredensial yang valid, penanganan data kosong, penanganan kredensial tidak sesuai, percobaan *login* berulang yang gagal, sensitivitas huruf pada *password*, ketahanan terhadap karakter khusus atau SQL injection, dan penanganan sesi kedaluwarsa. Hasil pengujian dapat dilihat pada tabel berikut

Tabel 4. 1 *Test case* pengguna melakukan *login*

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	<i>Login</i> dengan <i>username</i> dan <i>password</i> yang valid	<i>Username</i> = staff01 <i>Password</i> = Pass@123	Sistem berhasil melakukan autentikasi dan mengarahkan pengguna ke halaman <i>dashboard</i> sesuai hak akses	Berhasil masuk ke halaman sistem sesuai role pengguna	Valid
TC02	<i>Login</i> dengan <i>username</i> atau <i>password</i> kosong	<i>Username</i> = (kosong) <i>Password</i> = Pass@123 atau <i>Username</i> = staff01 <i>Password</i> = (kosong)	Sistem menolak proses <i>login</i> dan menampilkan pesan bahwa data harus diisi	Sistem menampilkan pesan validasi pada kolom yang kosong	Valid
TC03	<i>Login</i> dengan <i>username</i> atau <i>password</i> yang tidak sesuai	<i>Username</i> = staff999 <i>Password</i> = wrongpass	Sistem menolak proses <i>login</i> dan menampilkan pesan kesalahan autentikasi	Sistem menampilkan pesan bahwa <i>username</i> atau <i>password</i> salah	Valid
TC04	<i>Login</i> dengan <i>password</i> yang benar namun berbeda kapitalisasi huruf	<i>Username</i> = staff01 <i>Password</i> = pass@123 (huruf kecil)	Sistem menolak <i>login</i> karena <i>password</i> bersifat <i>case-sensitive</i>	Sistem menolak <i>login</i> dan menampilkan pesan kredensial salah	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $4/4 = 100\%$
 2. Persentase *test case* yang gagal = $0/4 = 0\%$
 3. Persentase *test case* yang telah diuji = $4/4 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/4 = 0\%$
- Jumlah total *test case* dalam pengujian = 4

b. *Test case* pengguna melakukan *logout*

Pengujian *logout* sistem dilakukan untuk memastikan bahwa pengguna benar-benar keluar dari sistem. Pengujian ini juga memastikan bahwa akses ke

halaman sistem setelah *logout* dibatasi dengan baik, termasuk pada kondisi sesi ganda di beberapa tab atau perangkat. Hasil pengujian ini bisa dilihat di tabel berikut.

Tabel 4. 2 *Test case* pengguna melakukan *logout*

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Melakukan <i>logout</i> dari halaman sistem	Pengguna menekan tombol <i>Logout</i>	Sistem berhasil mengakhiri sesi pengguna dan mengarahkan ke halaman <i>login</i>	Pengguna berhasil keluar dari sistem dan halaman <i>login</i> berhasil ditampilkan	Valid
TC02	Membatalkan konfirmasi <i>logout</i>	Pengguna menekan <i>Logout</i> lalu memilih Batal pada dialog konfirmasi	Sistem tetap berada di halaman sebelumnya dan sesi tidak berakhir	Sistem kembali ke halaman sebelumnya tanpa mengakhiri sesi	Valid
TC03	Mengakses kembali halaman sistem setelah <i>logout</i> menggunakan URL sebelumnya	Pengguna memasukkan URL halaman <i>dashboard</i> setelah <i>logout</i>	Sistem menolak akses dan mengarahkan pengguna kembali ke halaman <i>login</i>	Sistem berhasil mengarahkan pengguna ke halaman <i>login</i>	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $3/3 = 100\%$
2. Persentase *test case* yang gagal = $0/3 = 0\%$
3. Persentase *test case* yang telah diuji = $3/3 = 100\%$
4. Persentase *test case* yang belum diuji = $0/3 = 0\%$

Jumlah total *test case* dalam pengujian = 3

c. *Test case* Staf membuat pengajuan SPBy atau SPM LS

Pengujian *Black Box* pada fitur pembuatan pengajuan SPBy atau SPM LS dilakukan untuk memastikan bahwa staf dapat membuat pengajuan dengan benar. Dalam proses ini, sistem melakukan validasi data wajib serta validasi dokumen pendukung yang diunggah, termasuk format file dan ukuran maksimum yang diizinkan. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 3 *Test case* Staf membuat pengajuan SPBy atau SPM LS

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Membuat pengajuan dengan data lengkap dan	Semua field wajib diisi, dokumen pendukung diunggah, klik	Pengajuan tersimpan di sistem dengan status 'Menunggu	Pengajuan berhasil tersimpan dan status berubah	Valid

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
	valid	Simpan & Ajukan	Verifikasi'	menjadi Menunggu Verifikasi	
TC02	Membuat pengajuan dengan data tidak lengkap	Field 'Kegiatan' dikosongkan, klik Simpan & Ajukan	Sistem menampilkan pesan error 'Lengkapi data sebelum mengirim pengajuan'	Sistem menampilkan pesan validasi dan pengajuan tidak tersimpan	Valid
TC03	Upload dokumen pendukung dengan format valid	Unggah file PDF berukuran valid	Dokumen berhasil diunggah dan tercantum pada formulir pengajuan	Dokumen berhasil diunggah dan ditampilkan pada daftar lampiran	Valid
TC04	Upload dokumen pendukung dengan format selain PDF	Unggah file berformat .docx atau .jpg	Sistem menolak file dan menampilkan pesan bahwa hanya format PDF yang diizinkan	Sistem menampilkan pesan validasi format file dan upload ditolak	Valid
TC05	Upload dokumen pendukung dengan ukuran file melebihi batas maksimum	Unggah file PDF dengan ukuran di atas batas maksimum yang ditentukan sistem	Sistem menolak upload dan menampilkan pesan bahwa ukuran file melebihi batas yang diizinkan	Sistem menampilkan pesan validasi ukuran file dan upload ditolak	Valid
TC06	Mengirim pengajuan tanpa mengunggah dokumen pendukung	Semua field teks diisi lengkap, namun dokumen pendukung tidak diunggah, klik Simpan & Ajukan	Sistem menampilkan pesan bahwa dokumen pendukung wajib dilampirkan	Sistem menampilkan pesan validasi dan pengajuan tidak tersimpan	Valid
TC07	Mengisi nominal pengajuan dengan nilai negatif atau nol	Nominal = - 500000 atau 0	Sistem menolak nilai dan menampilkan pesan bahwa nominal harus lebih besar dari nol	Sistem menampilkan pesan validasi nominal tidak valid	Valid
TC08	Mengisi nominal pengajuan dengan karakter non-numerik	Nominal = 'abc123'	Sistem menolak input dan menampilkan pesan bahwa nominal harus berupa angka	Sistem menampilkan pesan validasi format nominal	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $8/8 = 100\%$
2. Persentase *test case* yang gagal = $0/8 = 0\%$

3. Persentase *test case* yang telah diuji = $8/8 = 100\%$

4. Persentase *test case* yang belum diuji = $0/8 = 0\%$

Jumlah total *test case* dalam pengujian = 8

d. *Test case* Staf melihat status pengajuan SPBy atau SPM LS

Pengujian *Black Box* pada fitur melihat status pengajuan dilakukan untuk memastikan Staf dapat memantau perkembangan pengajuan yang telah dikirimkan, termasuk kemampuan menyaring data serta batasan akses terhadap data milik staf lain. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 4. 4 *Test case* Staf melihat status pengajuan SPBy atau SPM LS

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Melihat daftar pengajuan yang telah dibuat	Buka menu Status Pengajuan	Daftar pengajuan milik staf tampil beserta status terkini	Daftar pengajuan berhasil ditampilkan dengan informasi status	Valid
TC02	Membuka detail salah satu pengajuan	Klik salah satu baris pengajuan pada daftar	Sistem menampilkan detail status, catatan verifikasi, dan riwayat proses	Detail pengajuan berhasil ditampilkan dengan informasi lengkap	Valid
TC03	Tidak ada pengajuan yang pernah dibuat	Buka menu Status Pengajuan pada akun baru tanpa pengajuan	Sistem menampilkan pesan 'Belum ada pengajuan yang dibuat'	Pesan informasi kosong berhasil ditampilkan	Valid
TC04	Menyaring daftar pengajuan berdasarkan status tertentu	Pilih filter status = 'Diajukan'	Daftar hanya menampilkan pengajuan dengan status "Diajukan "	Daftar menampilkan pengajuan dengan status "Diajukan	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $4/4 = 100\%$

2. Persentase *test case* yang gagal = $0/4 = 0\%$

3. Persentase *test case* yang telah diuji = $4/4 = 100\%$

4. Persentase *test case* yang belum diuji = $0/4 = 0\%$

Jumlah total *test case* dalam pengujian = 4

e. *Test case* Kepala Unit mengelola jadwal anggaran dan entry progres fisik

Pengujian *Black Box* dilakukan pada fitur pengelolaan jadwal anggaran dan entry progres fisik. Tujuannya adalah untuk memastikan Kepala Unit dapat mengisi dan memperbarui data jadwal serta progres fisik kegiatan dengan benar. Pengujian ini juga memeriksa validasi terhadap tanggal yang tidak valid dan nilai progres yang berada di luar rentang yang wajar. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 5 *Test case* Kepala Unit mengelola jadwal anggaran dan entry progres fisik

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Input jadwal dan progres fisik dengan data valid	tanggal_mulai = 01/07/2025 tanggal_selesai = 31/07/2025 progres = 75%	Jadwal dan data progres fisik berhasil disimpan di sistem	Data jadwal dan progres fisik berhasil tersimpan	Valid
TC02	Input tanggal yang tidak valid (tanggal selesai lebih awal dari mulai)	tanggal_mulai = 31/07/2025 tanggal_selesai = 01/07/2025	Sistem menampilkan pesan kesalahan validasi tanggal	Pesan kesalahan validasi berhasil ditampilkan	Valid
TC03	Input nilai progres fisik melebihi 100%	progres = 110%	Sistem menampilkan peringatan dan meminta konfirmasi sebelum menyimpan	Sistem berhasil menampilkan peringatan nilai progres tidak valid	Valid
TC04	Input nilai progres fisik dengan angka negatif	progres = -10%	Sistem menolak input dan menampilkan pesan bahwa nilai progres tidak boleh negatif	Sistem menampilkan pesan validasi nilai progres tidak valid	Valid
TC05	Input nilai progres fisik dengan karakter non-numerik	progres = 'tujuh puluh'	Sistem menolak input dan menampilkan pesan bahwa progres harus berupa angka	Sistem menampilkan pesan validasi format input	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $5/5 = 100\%$
 2. Persentase *test case* yang gagal = $0/5 = 0\%$
 3. Persentase *test case* yang telah diuji = $5/5 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/5 = 0\%$
- Jumlah total *test case* dalam pengujian = 5

f. *Test case* Kepala Unit melakukan verifikasi pengajuan SPBy atau SPM LS

Pengujian *Black Box* pada fitur verifikasi pengajuan bertujuan untuk memastikan bahwa Kepala Unit dapat memeriksa dan memverifikasi atau mengembalikan pengajuan yang dikirimkan oleh Staf. Selain itu, pengujian ini juga memastikan sistem menangani kondisi daftar verifikasi kosong, verifikasi ganda, dan pembatasan akses verifikasi antar unit. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 6 *Test case* Kepala Unit melakukan verifikasi pengajuan SPBy atau SPM LS

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Verifikasi pengajuan dengan data lengkap	Pilih pengajuan berstatus 'Menunggu Verifikasi' → klik Verifikasi	Status berubah menjadi 'Terverifikasi' dan notifikasi dikirim ke PPK	Status pengajuan berhasil diperbarui dan notifikasi terkirim	Valid
TC02	Mengembalikan pengajuan dengan catatan	Pilih pengajuan → klik Kembalikan → isi catatan alasan → Simpan	Status berubah menjadi 'Dikembalikan' dan notifikasi dikirim ke Staf	Pengajuan berhasil dikembalikan dengan catatan dan notifikasi terkirim	Valid
TC03	Tidak ada pengajuan yang menunggu verifikasi	Buka menu Verifikasi Pengajuan	Sistem menampilkan pesan 'Tidak ada pengajuan menunggu verifikasi'	Daftar kosong berhasil ditampilkan sesuai kondisi	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $3/3 = 100\%$
2. Persentase *test case* yang gagal = $0/3 = 0\%$
3. Persentase *test case* yang telah diuji = $3/3 = 100\%$
4. Persentase *test case* yang belum diuji = $0/3 = 0\%$

Jumlah total *test case* dalam pengujian = 3

g. *Test case* menyetujui atau menolak pengajuan SPBy atau SPM LS

Pengujian *Black Box* pada fitur persetujuan pengajuan dilakukan untuk memastikan bahwa PPK, Bendahara, dan PPSPM dapat menyetujui atau menolak pengajuan yang telah diverifikasi. Ini termasuk validasi kolom alasan penolakan dan pembatasan aksi pada pengajuan yang belum memenuhi status yang sesuai. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 7 *Test case* menyetujui atau menolak pengajuan SPBy atau SPM LS

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Menyetujui pengajuan yang telah terverifikasi	Pilih pengajuan berstatus 'Terverifikasi' → klik Setujui	Status berubah menjadi 'Disetujui PPK' dan notifikasi dikirim ke Pejabat Penandatanganan SPM	Status pengajuan berhasil diperbarui dan notifikasi terkirim	Valid
TC02	Menolak pengajuan dengan alasan	Pilih pengajuan → klik Tolak → isi alasan penolakan → Simpan	Status berubah menjadi 'Ditolak' dan alasan penolakan tersimpan	Pengajuan berhasil ditolak dengan alasan tersimpan	Valid
TC03	Tidak ada pengajuan berstatus Terverifikasi	Buka menu Persetujuan Pengajuan	Sistem menampilkan pesan 'Tidak ada pengajuan untuk disetujui'	Daftar kosong berhasil ditampilkan sesuai kondisi	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $3/3 = 100\%$
2. Persentase *test case* yang gagal = $0/3 = 0\%$
3. Persentase *test case* yang telah diuji = $3/3 = 100\%$
4. Persentase *test case* yang belum diuji = $0/3 = 0\%$

Jumlah total *test case* dalam pengujian = 3

- h. *Test case* PPK memvalidasi dan menyetujui pengajuan yang telah diverifikasi

Pengujian *Black Box* dilakukan pada fitur validasi PPK. Tujuannya adalah untuk memastikan bahwa PPK dapat memberikan persetujuan akhir terhadap pengajuan yang memenuhi syarat. Selain itu, pengujian ini juga memastikan bahwa sistem membatasi akses pada pengajuan yang tidak sesuai dengan statusnya dan mencegah persetujuan berulang. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 8 *Test case* PPK memvalidasi dan menyetujui pengajuan yang telah diverifikasi

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Memvalidasi dan menyetujui pengajuan yang siap	Buka daftar → pilih pengajuan 'Menunggu Persetujuan PPK' → isi data → klik Setujui	Status pengajuan berubah menjadi 'Disetujui'	Status pengajuan berhasil diperbarui menjadi Disetujui	Valid
TC02	Mengakses	Coba buka	Sistem tidak	Akses ditolak	Valid

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
	pengajuan dengan status tidak sesuai	pengajuan berstatus 'Menunggu Verifikasi' dari menu PPK	menampilkan atau tidak mengizinkan aksi pada pengajuan tersebut	dan pengajuan tidak tampil di daftar PPK	

Hasil Analisis:

1. Persentase *test case* yang lolos = $2/2 = 100\%$
2. Persentase *test case* yang gagal = $0/2 = 0\%$
3. Persentase *test case* yang telah diuji = $2/2 = 100\%$
4. Persentase *test case* yang belum diuji = $0/2 = 0\%$

Jumlah total *test case* dalam pengujian = 2

- i. *Test case* Bendahara mencatat entry realisasi SPBy atau SPM

Pengujian *Black Box* dilakukan pada fitur entry realisasi untuk memastikan bahwa Bendahara dapat mencatat realisasi pembayaran dengan benar. Ini termasuk memeriksa validasi jumlah yang melebihi nilai pengajuan, kelengkapan data, serta penanganan input nominal yang tidak valid. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 9 *Test case* Bendahara mencatat entry realisasi SPBy atau SPM

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Mencatat realisasi dengan jumlah valid	Pilih pengajuan 'Disetujui' → isi jumlah realisasi valid → klik Simpan	Realisasi tersimpan, saldo anggaran diperbarui, status menjadi 'Terealisasi'	Data realisasi berhasil disimpan dan saldo anggaran diperbarui	Valid
TC02	Jumlah realisasi melebihi nilai pengajuan	Masukkan jumlah realisasi melebihi nilai pengajuan yang disetujui	Sistem menampilkan peringatan bahwa jumlah melebihi nilai pengajuan	Peringatan melebihi nilai pengajuan berhasil ditampilkan	Valid
TC03	Data realisasi tidak lengkap (field tanggal kosong)	Biarkan field tanggal realisasi kosong → klik Simpan	Sistem menampilkan pesan error dan data tidak tersimpan sebelum dilengkapi	Pesan validasi berhasil ditampilkan dan data tidak tersimpan	Valid
TC04	Mengisi jumlah realisasi dengan nilai negatif	jumlah realisasi = -100000	Sistem menolak input dan menampilkan pesan bahwa jumlah realisasi harus lebih besar dari nol	Sistem menampilkan pesan validasi nominal tidak valid	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $4/4 = 100\%$

2. Persentase *test case* yang gagal = $0/4 = 0\%$
 3. Persentase *test case* yang telah diuji = $4/4 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/4 = 0\%$
- Jumlah total *test case* dalam pengujian = 4

j. *Test case* mencetak dokumen pengajuan yang disetujui

Pengujian pada fitur cetak dokumen juga dilakukan untuk memastikan bahwa pengguna dapat mencetak atau menyimpan dokumen pengajuan yang sudah disetujui dalam format PDF. Selain itu, pengujian ini juga memastikan bahwa opsi cetak tidak tersedia untuk pengajuan yang belum final serta konsistensi data pada dokumen yang dihasilkan. Hasil pengujian bisa dilihat pada tabel berikut.

Tabel 4. 10 *Test case* mencetak dokumen pengajuan yang disetujui

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Mencetak dokumen pengajuan berstatus 'Disetujui'	Pilih pengajuan disetujui → preview dokumen → klik Cetak	Dialog printer muncul dan dokumen dicetak	Dokumen berhasil dicetak sesuai format yang ditentukan	Valid
TC02	Menyimpan dokumen sebagai PDF	Pilih pengajuan disetujui → preview → klik Simpan sebagai PDF	File PDF berhasil diunduh ke perangkat pengguna	File PDF berhasil dibuat dan dapat diunduh	Valid
TC03	Mencetak dokumen pengajuan berstatus 'Ditolak'	Coba akses menu cetak pada pengajuan yang berstatus Ditolak	Opsi cetak tidak tersedia atau tombol cetak tidak aktif	Tombol cetak tidak aktif untuk pengajuan berstatus Ditolak	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $3/3 = 100\%$
 2. Persentase *test case* yang gagal = $0/3 = 0\%$
 3. Persentase *test case* yang telah diuji = $3/3 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/3 = 0\%$
- Jumlah total *test case* dalam pengujian = 3

k. *Test case* KPA mengelola master data anggaran

Pengujian *Black Box* pada fitur pengelolaan master data anggaran bertujuan memastikan KPA bisa menambahkan, mengedit, dan menghapus data anggaran. Pengujian ini juga memverifikasi kode anggaran duplikat serta validasi nominal dan penanganan data yang sudah memiliki keterkaitan transaksi. Hasilnya bisa dilihat di tabel berikut

Tabel 4. 11 *Test case* KPA mengelola master data anggaran

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Menambah data anggaran baru	Isi kode anggaran baru dan nama kegiatan → klik Simpan	Data baru tersimpan dan tampil di daftar master anggaran	Data anggaran berhasil ditambahkan dan ditampilkan	Valid
TC02	Mengubah data anggaran yang sudah ada	Edit nama kegiatan yang ada → klik Simpan	Perubahan data berhasil disimpan dan ditampilkan	Data anggaran berhasil diperbarui sesuai perubahan	Valid
TC03	Menambah dengan kode anggaran yang sudah terdaftar (duplikat)	Masukkan kode anggaran yang sudah ada → klik Simpan	Sistem menampilkan pesan 'Kode anggaran sudah digunakan'	Pesan validasi duplikat berhasil ditampilkan	Valid
TC04	Menghapus data anggaran	Pilih data → klik Hapus → konfirmasi penghapusan	Data terhapus setelah konfirmasi; tidak dapat dibatalkan	Data anggaran berhasil dihapus setelah konfirmasi	Valid
TC05	Mengisi nilai pagu anggaran dengan angka negatif atau non-numerik	nilai pagu = -1000000 atau 'sejuta'	Sistem menolak input dan menampilkan pesan bahwa nilai pagu harus berupa angka positif	Sistem menampilkan pesan validasi nominal tidak valid	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $5/5 = 100\%$
2. Persentase *test case* yang gagal = $0/5 = 0\%$
3. Persentase *test case* yang telah diuji = $5/5 = 100\%$
4. Persentase *test case* yang belum diuji = $0/5 = 0\%$

Jumlah total *test case* dalam pengujian = 5

1. *Test case* KPA melihat semua data anggaran dan realisasi

Pengujian *Black Box* dilakukan untuk fitur melihat data anggaran dan realisasi. Tujuannya adalah memastikan KPA dapat memantau *dashboard* ringkasan anggaran dengan baik. KPA juga harus bisa menggunakan filter data, kombinasi filter, dan mengekspor data sesuai kebutuhan mereka. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 4. 12 *Test case* KPA melihat semua data anggaran dan realisasi

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Melihat <i>dashboard</i> ringkasan anggaran	Buka menu Data Anggaran & Realisasi	<i>Dashboard</i> menampilkan total pagu, terpakai, dan sisa anggaran	<i>Dashboard</i> ringkasan anggaran berhasil ditampilkan	Valid
TC02	Filter data berdasarkan tahun dan unit	Pilih tahun = 2025, unit = TI → terapkan filter	Tabel dan grafik menampilkan data sesuai filter yang dipilih	Data berhasil difilter dan ditampilkan sesuai parameter	Valid
TC03	Ekspor data anggaran ke Excel	Klik tombol Ekspor → pilih format Excel	File Excel berhasil diunduh ke perangkat pengguna	File Excel berhasil dibuat dan dapat diunduh	Valid
TC04	Filter dengan kondisi data tidak ditemukan	Pilih tahun yang tidak memiliki data anggaran	Sistem menampilkan pesan 'Data tidak ditemukan'	Pesan data tidak ditemukan berhasil ditampilkan	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $4/4 = 100\%$
 2. Persentase *test case* yang gagal = $0/4 = 0\%$
 3. Persentase *test case* yang telah diuji = $4/4 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/4 = 0\%$
- Jumlah total *test case* dalam pengujian = 4

m. *Test case* KPA mengakses laporan kegiatan anggaran

Pengujian *Black Box* pada fitur laporan kegiatan anggaran dilakukan untuk memastikan KPA dapat mengakses laporan realisasi dengan berbagai parameter dan mengunduh laporan dalam format PDF maupun Excel, termasuk penanganan parameter rentang tanggal yang tidak valid. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 4. 13 *Test case* KPA mengakses laporan kegiatan anggaran

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Mengakses laporan realisasi dengan parameter lengkap	Pilih jenis laporan 'Realisasi' → atur parameter → klik Tampilkan	Laporan tampil dengan data sesuai parameter yang ditentukan	Laporan kegiatan anggaran berhasil ditampilkan	Valid
TC02	Mengunduh laporan dalam format PDF	Klik Unduh → pilih format PDF	File PDF laporan berhasil diunduh	File PDF berhasil dibuat dan dapat diunduh	Valid
TC03	Mengunduh laporan dalam format Excel	Klik Unduh → pilih format Excel	File Excel laporan berhasil diunduh	File Excel berhasil dibuat dan dapat diunduh	Valid
TC04	Mengakses laporan dengan rentang tanggal tidak valid (tanggal akhir lebih awal dari tanggal awal)	tanggal_awal = 31/12/2025 tanggal_akhir = 01/01/2025	Sistem menampilkan data tidak ada	Data tidak ada	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $4/4 = 100\%$
2. Persentase *test case* yang gagal = $0/4 = 0\%$
3. Persentase *test case* yang telah diuji = $4/4 = 100\%$
4. Persentase *test case* yang belum diuji = $0/4 = 0\%$

Jumlah total *test case* dalam pengujian = 4

n. *Test case* KPA mengelola mapping pagu unit dan PPK ke akun

Pengujian *Black Box* untuk fitur mapping pagu unit dan PPK ke akun dilakukan untuk memastikan KPA dapat mengatur pemetaan anggaran dengan benar. Validasi ini penting untuk memeriksa apakah total alokasi anggaran melebihi pagu yang telah ditentukan, serta mencegah duplikasi mapping dan nilai pagu yang tidak wajar. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 14 *Test case* KPA mengelola mapping pagu unit dan PPK ke akun

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Menambah mapping baru dengan alokasi valid	Isi unit, PPK, akun, dan nilai pagu → klik Simpan	Mapping tersimpan dan tampil di daftar konfigurasi	Mapping pagu berhasil ditambahkan dan ditampilkan	Valid
TC02	Total alokasi	Masukkan nilai	Sistem	Peringatan	Valid

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
	melebihi pagu akun yang tersedia	pagu sehingga total alokasi melebihi pagu akun	menampilkan peringatan bahwa total alokasi melebihi pagu	melebihi pagu berhasil ditampilkan	
TC03	Menghapus mapping yang sudah ada	Pilih mapping → klik Hapus → konfirmasi	Mapping terhapus setelah konfirmasi penghapusan	Mapping berhasil dihapus setelah konfirmasi	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $3/3 = 100\%$
2. Persentase *test case* yang gagal = $0/3 = 0\%$
3. Persentase *test case* yang telah diuji = $3/3 = 100\%$
4. Persentase *test case* yang belum diuji = $0/3 = 0\%$

Jumlah total *test case* dalam pengujian = 3

o. *Test case* KPA mengelola tahun anggaran

Pengujian *Black Box* pada fitur pengelolaan tahun anggaran dilakukan untuk memastikan KPA dapat menambah tahun anggaran baru, menutup periode aktif, serta sistem menangani duplikasi tahun anggaran dan konsistensi data saat suatu periode sudah memiliki transaksi. Hasil pengujian dapat dilihat pada tabel berikut

Tabel 4. 15 *Test case* KPA mengelola tahun anggaran

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Menambah tahun anggaran baru	Isi tahun = 2026 → klik Simpan	Tahun anggaran 2026 berhasil ditambahkan ke dalam sistem	Tahun anggaran berhasil ditambahkan dan tampil di daftar	Valid
TC02	Menutup periode tahun anggaran aktif	Ubah status tahun aktif → pilih Tutup → konfirmasi	Sistem meminta konfirmasi bahwa aksi tidak dapat dibatalkan, lalu periode ditutup	Tahun anggaran berhasil ditutup setelah konfirmasi	Valid
TC03	Menambah tahun anggaran yang sudah terdaftar (duplikat)	Isi tahun = 2025 (sudah ada) → klik Simpan	Sistem menampilkan pesan 'Tahun anggaran sudah terdaftar'	Pesan validasi duplikat berhasil ditampilkan	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $3/3 = 100\%$

2. Persentase *test case* yang gagal = $0/3 = 0\%$
 3. Persentase *test case* yang telah diuji = $3/3 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/3 = 0\%$
- Jumlah total *test case* dalam pengujian = 3

p. *Test case* KPA mengelola unit dan user

Pengujian *Black Box* pada fitur pengelolaan unit dan user dilakukan untuk memastikan KPA dapat menambah, mengubah, dan menonaktifkan user. Pengujian ini juga memastikan KPA dapat memvalidasi keunikan *username* dan *email*, akses *login* bagi user nonaktif dapat diblokir, serta penanganan perubahan role pada user yang sedang aktif. Hasil dari pengujian ini dapat dilihat pada tabel berikut.

Tabel 4. 16 *Test case* KPA mengelola unit dan user

No	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Status
TC01	Menambah user baru dengan data unik	<i>username</i> = user_baru <i>email</i> = baru@polibatam.ac.id role = Staf → klik Simpan	Akun user berhasil dibuat dan tampil pada daftar pengguna	User berhasil ditambahkan dan ditampilkan pada tabel	Valid
TC02	Menambah user dengan <i>username</i> yang sudah terdaftar	<i>username</i> = staff01 (sudah ada) → klik Simpan	Sistem menampilkan pesan ' <i>Username</i> sudah terdaftar'	Pesan validasi <i>username</i> duplikat berhasil ditampilkan	Valid
TC03	Menambah user dengan <i>email</i> yang sudah terdaftar	<i>email</i> = ada@polibatam.ac.id (sudah ada) → klik Simpan	Sistem menampilkan pesan ' <i>Email</i> sudah digunakan'	Pesan validasi <i>email</i> duplikat berhasil ditampilkan	Valid
TC04	Menonaktifkan user yang sedang aktif	Pilih user aktif → klik Nonaktifkan → konfirmasi	User dinonaktifkan dan akses <i>login</i> diblokir oleh sistem	User berhasil dinonaktifkan dan tidak dapat <i>login</i>	Valid

Hasil Analisis:

1. Persentase *test case* yang lolos = $4/4 = 100\%$
 2. Persentase *test case* yang gagal = $0/4 = 0\%$
 3. Persentase *test case* yang telah diuji = $4/4 = 100\%$
 4. Persentase *test case* yang belum diuji = $0/4 = 0\%$
- Jumlah total *test case* dalam pengujian = 4

BAB V KESIMPULAN

5.1 Kesimpulan

Proyek Akhir ini telah berhasil merancang dan membangun ulang Sistem Informasi Pengelolaan Anggaran di Politeknik Negeri Batam, yang mencakup tahap perencanaan (penetapan master data dan pagu anggaran), pengajuan, verifikasi berjenjang, validasi, realisasi, hingga pelaporan anggaran. Proyek ini melakukan migrasi total dari platform desktop yang menggunakan C# ke aplikasi web yang menggunakan PHP dengan Framework Laravel. Sistem Informasi Pengelolaan Anggaran yang baru ini berhasil mengimplementasikan enam belas kebutuhan fungsional. Kebutuhan fungsional tersebut mencakup seluruh alur administrasi keuangan, seperti pengajuan, verifikasi berjenjang, validasi, dan pelaporan anggaran. Sistem Informasi Pengelolaan Anggaran ini juga telah berhasil terintegrasi dengan sistem *Single Sign-On internal* kampus Polibatam.

Selain itu, penggunaan basis data PostgreSQL menjaga integritas data dengan memodelkan struktur hierarki anggaran yang teratur mengikuti standar RKAKL. Hasil pengujian *User Acceptance Testing* bersama para pemangku kepentingan menunjukkan bahwa Sistem Informasi Pengelolaan Anggaran yang dihasilkan lebih stabil, responsif, dan minim kesalahan teknis. Sistem Informasi Pengelolaan Anggaran ini juga siap untuk dikelola secara mandiri oleh tim IT *internal*. Dengan demikian, ketergantungan terhadap pihak pengembang *eksternal* dapat dikurangi.

5.2 Saran

Untuk pengembangan sistem lebih lanjut agar cakupan fungsionalitas dan dampaknya menjadi lebih luas di masa mendatang, beberapa saran yang dapat direkomendasikan adalah:

1. Pengembangan Fitur Tambahan

Sistem dapat dilengkapi dengan fitur notifikasi otomatis melalui *email* atau *WhatsApp Gateway* untuk memberikan informasi kepada Kepala Unit atau PPK terkait dokumen yang memerlukan verifikasi atau persetujuan.

2. Peningkatan Keamanan Sistem

Pengujian keamanan, seperti *penetration testing*, perlu dilakukan secara berkala untuk mendeteksi potensi celah keamanan seiring dengan perkembangan sistem dan pembaruan framework Laravel.

3. Perluasan Integrasi Sistem

Integrasi yang saat ini terbatas pada lingkungan *internal* Polibatam dapat dikembangkan lebih lanjut, misalnya dengan sistem perbankan untuk mendukung proses pencairan dana secara lebih cepat.

4. Pelatihan Pengguna

Pelatihan bagi staf administrasi dan pihak terkait perlu dilakukan untuk membantu proses adaptasi terhadap penggunaan sistem berbasis web yang baru.

DAFTAR PUSTAKA

- Ariyanto, Y., Farhan, M., Rachmad, F., & Puspitasari, D. (2024). Issue 2 Year 2024 Pages 66-73 Jurnal Manajemen Teknologi dan Informatika. *Matrix: Jurnal Manajemen Teknologi Dan Informatika*, 14(2), 66–73.
- Effendi. (n.d.). Pengelolaan Perguruan Tinggi Negeri Badan. *Evaluasi Pengelolaan PTNBH*, 2023.
- Hogan, G., Shalkauskaite, P., Zhu, M., Derwin, M., Yilmaz, M., McCarren, A., & Clarke, P. M. (2024). Investigating Systems Modernisation: Approaches, Challenges and Risks. *Communications in Computer and Information Science*, 2179 CCIS, 147–162. https://doi.org/10.1007/978-3-031-71139-8_10
- Installation | Laravel 13.x - The clean stack for Artisans and agents*. (n.d.). Retrieved June 17, 2026, from <https://laravel.com/docs/13.x>
- Irawan, A., & Devitra, J. (2020). Analisis Dan Perancangan Sistem Informasi Manajemen Anggaran Pada Dinas Pendidikan Provinsi Jambi. *Jurnal Manajemen Sistem ...*, 5(3), 395–405. <http://ejournal.stikom-db.ac.id/index.php/manajemensisteminformasi/article/view/913>
- MVC (Model, View, Controller) - BINUS @Bekasi - Kampus Beken Asyik | Business Service and Technology*. (n.d.). Retrieved June 17, 2026, from <https://binus.ac.id/bekasi/2025/07/mvc-model-view-controller/>
- Octavian Endra Saputra, & Dendy Kurniawan. (2024). Sistem Informasi Akuntansi Perencanaan Anggaran Biaya Kontruksi Pada Cv. Teguh Sejahtera Dengan Metode Activity Based Budgeting Tahun 2023. *Jurnal Riset Sistem Informasi*, 1(2), 01–09. <https://doi.org/10.69714/ejj5sn79>
- PHP: Preface - Manual*. (n.d.). Retrieved June 17, 2026, from <https://www.php.net/manual/en/>
- PostgreSQL: Documentation*. (n.d.). Retrieved June 17, 2026, from <https://www.postgresql.org/docs/>
- Sallu, S., Harsono, Y., & Fajarianto, O. (2023). Implementation of *Waterfall* Method in Model Development to Improve Learning Quality of Computer Network Courses. *JTP - Jurnal Teknologi Pendidikan*, 25(3), 496–513.

<https://doi.org/10.21009/jtp.v25i3.44418>

- Sistem Informasi Transparansi Anggaran dan Pendanaan Berbasis Web dalam Mendukung Tata Kelola Keuangan Kampus Oleh, P., El Rezen Purba, D., Limbong, T., Torikul Huda, A., & Pinem, S. (2025). Pengembangan Sistem Informasi Transparansi Anggaran dan Pendanaan (St Anna) Berbasis Web dalam Mendukung Tata Kelola Keuangan Kampus. *JUKI : Jurnal Komputer Dan Informatika*, 7(2), 228–239. <https://doi.org/10.53842/JUKI.V7I2.1910>
- Sistem, P., Di, A., Perencanaan, B., Daerah, P., Bappeda, ", Kantor, ", Medan, W., Web, B., Saputra, Y., Dhany, H. W., Perwitasari, I. D., Pembangunan, U., & Medan, P. (2024). Penerapan Sistem Anggaran Di Badan Perencanaan Pembangunan Daerah “BAPPEDA” Kantor Walikota Medan Berbasis Web. *Jurnal Minfo Polgan*, 13(2), 1897–1905. <https://doi.org/10.33395/JMP.V13I2.14316>
- Suherdiansyah, F., & Devitra, J. (2020). Analisis dan Perancangan Sistem Informasi Manajemen Anggaran Pada Badan Perencanaan Pembangunan Daerah Provinsi Jambi. *Jurnal Manajemen Sistem Informasi*, 5(2), 300–312.
- Suryani, N. K., Widya Utami, N., & Grana Aristyana Dewi, E. (2024). Analisis Dan Perancangan Sistem Informasi Perencanaan Dan Realisasi Anggaran Dana Desa Di Desa Belok/ Sidan. *Teknologi*, 14(2), 72–84. <https://doi.org/10.26594/teknologi.v14i2.4863>

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

Dokumentasi



Wawancara

https://polnebat-my.sharepoint.com/:f/g/personal/firjatullah_3312311111_students_polibatam_ac_id/IgDAVjXZlxNPSbIzITAshvkRAZooi7TJ_M48IHv6930YBSc?e=xgMUNg

Lampiran B. Link Produk

<https://github.com/riadyagus1/SIAP>

Lampiran C. Dokumen Pengujian

https://polnebat-my.sharepoint.com/:f/g/personal/firjatullah_3312311111_students_polibatam_ac_id/IgCLDz57ScU4QIolYjCgf7fMAWeGWrNSo7Xb9yXyexxrewE?e=6DTgeU

Lampiran D. Demo

<https://polnebat->

[my.sharepoint.com/:f/g/personal/firjatullah_3312311111_students_polibatam_ac_id/IgAG5Q_rUVLASbfb1TQSLQ7PAbhbpwv2naNa-U7TcwDGE?e=9bY1FI](https://polnebat-my.sharepoint.com/:f/g/personal/firjatullah_3312311111_students_polibatam_ac_id/IgAG5Q_rUVLASbfb1TQSLQ7PAbhbpwv2naNa-U7TcwDGE?e=9bY1FI)