BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam program magang STARTECH yang di selenggarakan oleh Astra Credit Companies, kedudukan yang di berikan adalah *internee* di *section* IT *Security* di bawah naungan departemen *IT Service & Security* di *Head Office Astra Credit Companies*, Fatmawati. Sebagai *internee* IT *Security* akan aktif untuk melakukan koordinasi kepada beberapa orang di antaranya kepada:

1. **Kepala Departemen**, Martin Prima Asril selaku *department head* untuk melakukan *mapping* atau mempelajari tentang IT *Service & Security*, terkait cara kerja *service* & langkah perlindungan untuk keamanan *mobile* atau *web app* di Astra Credit Companies, & terkait tugas akhir magang.
2. **Mentor Departemen**, Jessy Ratna Wulandari selaku mentor departemen untuk berkoordinasi terkait tugas akhir yang dilakukan, arahan untuk membantu departemen, dan terkait halaman presentasi untuk tugas akhir dari program magang ini.
3. **Human Capital**, Marcelia Debby selaku mentor *Human Capital* (HC) atau mentor dari program STARTECH yang aktif berkoordinasi terkait kesulitan dan hambatan selama menjalankan program magang STARTECH, kriteria tugas akhir, progres tugas akhir dan juga terkait *event* yang di laksanakan selama menjalankan magang STARTECH.

3.2 Pembelajaran Selama Magang

Selama menjalani program magang di bagian IT Security, pembelajaran penting yang didapat mencakup aspek teknis, prosedural, dan operasional. Pembelajaran diperoleh dari kolaborasi lintas tim, baik secara langsung dalam tugas maupun melalui diskusi dan bimbingan. Berikut ini merupakan rincian pembelajaran selama magang:

3.2.1 Tim *IT Governance*

Tim ini berfokus pada pengelolaan kebijakan, standar, dan tata kelola TI yang sesuai dengan regulasi dan kebutuhan organisasi.

- Memahami alur proses *audit* internal, termasuk tahapan penyusunan, peninjauan ulang, dan pengesahan dokumen kebijakan IT.
- Mempelajari persiapan teknis dan administratif menjelang pelaksanaan *audit*, seperti kelengkapan dokumen, bukti implementasi, serta tanggung jawab masing-masing unit kerja.
- Mengamati secara langsung jalannya proses *audit ISO 27001* yang sedang berlangsung selama periode magang, termasuk interaksi dengan auditor eksternal dan penyesuaian terhadap temuan audit.
- Terlibat dalam proses *review* dan *update* terhadap dokumen kebijakan yang harus disesuaikan dengan kondisi operasional dan sistem keamanan terkini.

3.2.2 Tim *IT Security Operation*

Tim ini menangani operasional teknis terkait keamanan sistem, termasuk simulasi serangan dan pengujian fitur baru.

- Melaksanakan *penetration testing* sederhana pada fitur *OTP* (One-Time Password) menggunakan sistem operasi Linux sebagai bagian dari pengujian keamanan aplikasi yang sedang dikembangkan.
- Menjalankan simulasi serangan *brute force* dengan teknik *intruder request* secara terus-menerus untuk menguji ketahanan sistem terhadap potensi serangan *DDoS via web application*.
- Mengidentifikasi kemungkinan *bypass OTP verification* dengan cara memodifikasi *HTTP response* dari sisi klien, lalu meneruskannya ke server untuk melihat apakah validasi OTP bisa dilewati tanpa proses autentikasi yang sah.
- Mempelajari pemanfaatan *crypto.js* dalam proses enkripsi data *credential* seperti nomor telepon, email, dan NIK. Menyadari pentingnya menjaga agar proses *encrypt-decrypt* tidak terekspos di sisi klien karena dapat menimbulkan risiko kebocoran data.

- Mengenal teknik *code obfuscation* menggunakan *obfuscator* untuk mengacak *source code* agar lebih sulit dibaca dan dianalisis oleh pihak tidak berwenang.

3.2.3 Tim *IT Security Analyst*

Tim ini berfokus pada analisis kerentanan, pengelolaan risiko, serta penerapan otomatisasi keamanan melalui alat bantu modern.

- Menggunakan *PowerSploit* untuk melakukan pencarian file berisi data sensitif di penyimpanan jaringan internal seperti *ADNAS (Active Directory Network Attached Storage)*, yang digunakan untuk menyimpan dokumen penting seperti memo, KTP, dan informasi pribadi lainnya.
- Memahami penggunaan *SonarQube* sebagai alat *static application security testing (SAST)* dalam menganalisis kerentanan pada *source code* aplikasi secara otomatis tanpa perlu dijalankan terlebih dahulu.
- Mempelajari penggunaan *Nexus* sebagai platform manajemen kerentanan yang dapat mendeteksi, menampilkan, dan merekomendasikan tindakan perbaikan terhadap celah keamanan dalam aplikasi.
- Mengenal metode *wardriving*, yaitu teknik survei jaringan nirkabel untuk mendeteksi jumlah *access point* yang aktif, mengevaluasi tingkat enkripsinya, serta mengidentifikasi perangkat tak dikenal yang terhubung ke jaringan.
- Mempelajari prinsip dan praktik *DevSecOps*, yaitu integrasi antara tim pengembang (*developer*), keamanan (*security*), dan operasional (*operation*) guna mempercepat proses pengembangan sekaligus memastikan keamanan sejak awal.
- Mengenal penerapan *CI/CD pipeline* menggunakan *GitLab* dan *Jenkins* dalam proses otomatisasi pembangunan dan pengujian aplikasi. Mempelajari pula konsep *quality gate*, *branching strategy*, serta manajemen versi pada pengembangan perangkat lunak.

3.2.4 Tim *Security Operation Center (SOC)*

Tim SOC bertanggung jawab atas deteksi, analisis, dan respons terhadap insiden keamanan informasi secara menyeluruh dan real-time.

- Melakukan pemantauan keamanan siber secara menyeluruh selama 24/7 menggunakan platform *MxDR (Managed Extended Detection and Response)* serta solusi *EDR (Endpoint Detection and Response)* untuk mendeteksi dan merespons aktivitas mencurigakan pada perangkat pengguna akhir.
- Menganalisis kasus nyata di mana perangkat lunak seperti *Microsoft Word* memicu eksekusi *Command Prompt* atau *PowerShell*, yang kemudian menjalankan perintah berbahaya. Hal ini dianalisis melalui *log* EDR untuk mendeteksi potensi koneksi ke *domain* dengan reputasi buruk atau upaya koneksi ke perangkat lain.
- Mengenal dan memanfaatkan berbagai konsep dan alat bantu keamanan seperti:
  - *Crowdstrike*, *SIEM*, *PowerSploit*, dan *WAF (Web Application Firewall)*.
  - *CVSS (Common Vulnerability Scoring System)*: sistem skor kerentanan berbasis aspek kerahasiaan, integritas, dan ketersediaan.
  - *VPR (Vulnerability Priority Rating)*: penilaian prioritas penanganan kerentanan berdasarkan ancaman terkini.
  - *Threat intelligence*: pendekatan proaktif dalam mengidentifikasi dan memitigasi ancaman yang sedang tren.
  - *APT (Advanced Persistent Threat)*: jenis serangan siber yang terencana dan bertarget dalam jangka panjang.
- Memahami struktur organisasi SOC di perusahaan yang terdiri dari:
  - **Tier 1 (L1)**: bagian yang bertugas untuk melakukan pemantauan awal, investigasi dasar, dan penyelesaian tiket insiden.
  - **Tier 2 (L2)**: unit yang menangani respons lanjutan terhadap insiden serta terlibat langsung dengan *host* yang terdampak.
  - **Tier 3 (L3)**: tim yang berfokus pada analisis mendalam, termasuk *malware analysis*, *reverse engineering*, dan *digital forensic*.
- Mempelajari proses *network incident response* yang melibatkan identifikasi, isolasi, serta pemulihan jaringan yang terdampak insiden untuk mencegah penyebaran lebih lanjut.

3.3 Tugas yang Dilakukan

Tugas yang dilakukan selama menjalankan program magang sebagai *internee* adalah membantu pekerjaan di *section* dan melakukan sebuah *improvement* dari proses atau sistem kerja yang sudah ada di *section* atau departemen. Selain itu, dilakukan pula proses *review* dan *update* terhadap dokumen keamanan IT Security, mengingat masih terdapat banyak dokumen yang belum diperbarui atau belum sesuai dengan kondisi operasional saat ini.

Selama magang, tidak selalu terdapat pembagian tugas secara langsung atau spesifik. Oleh karena itu, dilakukan inisiatif untuk terlibat dalam aktivitas yang sedang berlangsung, seperti mempelajari alat atau aplikasi yang sedang di gunakan oleh *officer* dan pengelolaan bukti implementasi kebijakan keamanan, mendukung persiapan audit internal, serta mengamati dan mempelajari alur kerja masing-masing bagian di dalam tim IT Security. Dengan cara tersebut, pemahaman mengenai operasional keamanan informasi dapat diperoleh melalui pengalaman langsung dan keterlibatan aktif dalam kegiatan harian tim.

3.4 Uraian Pelaksanaan Magang

Program magang yang dilaksanakan, banyak hal yang dilakukan mulai dari pekerjaan dasar dan mempelajari hal baru yang sesuai dengan IT *Security* seperti memeriksa dokumen *mandatory*, mempelajari alur proses berjalannya audit dan lainnya. Untuk pelaksanaan kerja magang secara lengkap di jelaskan pada Tabel 3.1.

Tabel 3.1. Rincian pelaksanaan

Minggu Ke -	Pekerjaan yang Dilakukan
1	Melakukan perkenalan dengan super visi, pembimbing magang ACC, *office tour*, melakukan *session in class training*, mempelajari *skill* atau kemampuan basic yang di ajarkan langsung oleh mentor ACC, di antaranya *Basic Excel*, *Problem Solving*, *Google Data Studio*, *Presentation Skill*, *Improvement Ideas*.
	Lanjut pada halaman berikutnya

Tabel 3.1 Rincian pelaksanaan (lanjutan)

Minggu Ke -	**Pekerjaan yang Dilakukan**
2	Perkenalan departemen IT *Service & Security* untuk bagian apa saja di IT *Security, jobdesk,* dan perkenalan antar anggota tim (*Analyst, Operation, Support*, dan SOC).
3	Melakukan *mapping* departemen IT *Service & Security* dengan kepala departemen, dan mentor departemen mulai dari mengenali visi, misi dan peran IT *Service & Security* di ACC kemudian di lanjutkan dengan *mapping* tim *Analyst*, mempelajari *security awareness*, deteksi anomali Wi-Fi kantor HO ACC dan *security checking* pada aplikasi yang hendak di *publish*
4	*In class training* selama dua hari, mengikuti *sharing alumni session,* mendengar cerita suka & duka serta *demand* ke depannya dari program STARSHIP & STARTECH dan beberapa hari tersisa melakukan kembali *mapping* dengan tim SOC, mempelajari bagaimana cara mendeteksi serangan yang terpantau mengakses server, langkah, pencegahan dan respons yang di berikan terhadap ancaman serangan.
5	Melakukan kegiatan *core business* ACC yang mempelajari *section* lain, yaitu *Accountable Receivable* selama 3 hari di cabang ACC Kebon Jeruk dan kemudian mempersiapkan hasil pembelajaran untuk di presentasi dan sisa hari melakukan *mapping* dengan tim IT *Operation*
6	Mempelajari informasi terkait *section Accountable Receivable* dan mempersiapkan bahan presentasi di depan para senior yang telah di tetapkan jadi juri sebagai penilaian *skill* presentasi.
7	Mengenali departemen IT *Service & Security* dan berfokus untuk mencari sebuah problem untuk di *improvement* lewat *mapping* tim IT *Governance*, Kak Christin dan anggota tim *Analyst* Kak Oky

Lanjut pada halaman berikutnya

Tabel 3.1 Rincian pelaksanaan (lanjutan)

Minggu Ke -	**Pekerjaan yang Dilakukan**
8	*Mapping* dengan IT *Governance* untuk mengetahui bagaimana proses menjalani audit, data jumlah dokumen, informasi data belum di *update* dan sudah. Setelah itu melaporkan perkembangan dan konsultasi dengan mentor departemen dan mentor HC.
9	Setelah sudah mengetahui apa sebuah problem di departemen dan mulai menjalani sebuah aksi *improvement* yaitu memulai pembuatan *dashboard inventory* untuk *request* temuan atau hasil temuan yang ada.
10	Mengumpulkan data beserta informasi terkait permasalahan yang terjadi, kemudian di olah dan di persiapkan untuk pengajuan dan konsultasi dengan mentor departemen dan mentor HC.
11	Membuat halaman presentasi mengumpulkan data pendukung ataupun informasi yang di butuhkan untuk sebagai bahan presentasi mengapa harus melakukan improvisasi dan apa manfaatnya di masa yang akan datang.
12	Membuat *dashboard pentest inventory* tiga fitur awal yaitu *create, read*, dan *update*
13	Melanjutkan *dashboard pentest inventory* empat fitur selanjutnya yaitu *show open, show total, show closed* dan *show exclude*
14	Melanjutkan *dashboard pentest inventory* lima fitur selanjutnya yaitu *all, Mobile Android, Mobile IOS, Web* dan API
15	Melakukan *review* dokumen *mandatory* salah satunya dokumen kebijakan standar IT *Security*, mengumpulkan data dan informasi (berapa lama belum di *review* dan di-*update*)
16	Membuat *dashboard monitoring document* untuk memantau dokumen berdasarkan umur, di minggu ini membuat fitur *input* dan *update*

Lanjut pada halaman berikutnya

Tabel 3.1 Rincian pelaksanaan (lanjutan)

Minggu Ke -	Pekerjaan yang Dilakukan
17	Melanjutkan membuat *dashboard monitoring document* pada bagian visualisasi dengan menggunakan *chart* kemudian membuat fitur *sort document by date* atau umur dokumen (berumur satu, tiga, dan lima tahun)
18	Melakukan sosialisasi untuk *dashboard inventory pentest* dan *monitoring document*, untuk di gunakan oleh PIC yang bersangkutan dan mempresentasikan tugas akhir di depan tiga *departemen head* dari divisi lain

3.5 Project Magang

Hasil akhir dari program magang ini berupa *improvement* di *section* IT Security melalui pengembangan dua *dashboard*, yaitu *inventory penetration* dan *monitoring dokumen*. Keduanya dirancang untuk meningkatkan efisiensi kerja, mempercepat pelacakan permintaan pengujian keamanan, serta mendukung kesiapan dokumen dalam proses audit.

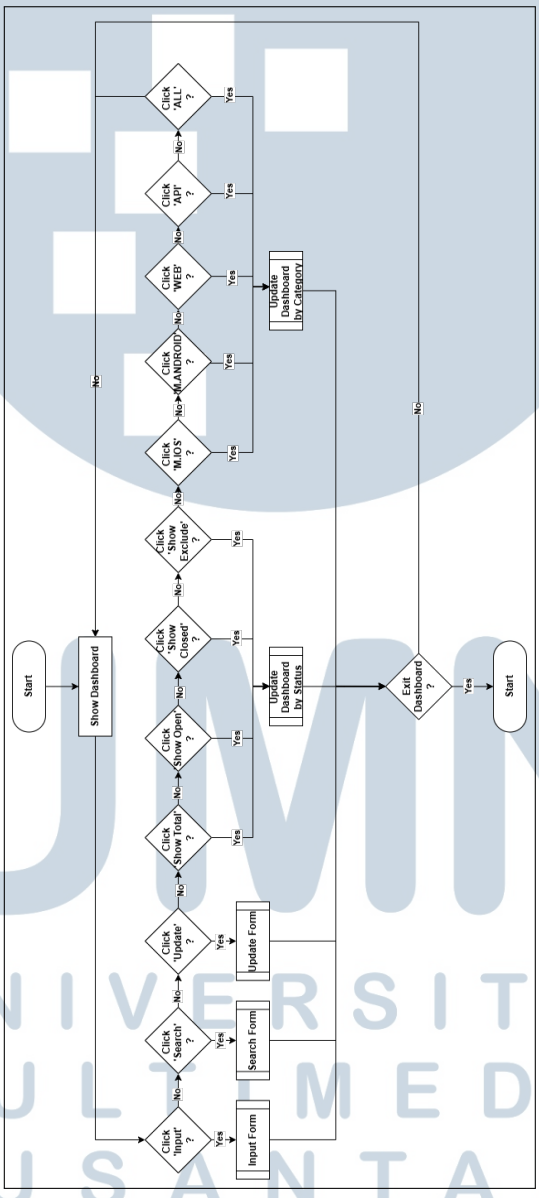
Kedua *dashboard* dikembangkan menggunakan *Microsoft Excel* dan *VBA* sesuai arahan mentor, sebagai solusi cepat, efektif, dan sesuai kebutuhan tim. Pemilihan Excel bertujuan menggantikan pencatatan manual dengan sistem yang lebih terstruktur, tanpa menambah biaya, infrastruktur, atau pelatihan karena sudah familiar digunakan oleh tim.

Alternatif solusi lain seperti pengembangan berbasis web atau sistem informasi terintegrasi tidak digunakan karena dinilai memerlukan waktu lebih lama, tidak sepraktis Excel dalam implementasi, serta dinilai terlalu kompleks untuk kebutuhan yang bersifat administratif dan operasional. Dashboard yang dibangun dilengkapi dengan fitur pendataan, status pembaruan, identifikasi tanggung jawab, serta indikator visual berbasis *conditional formatting* yang memudahkan pemantauan secara langsung.

Proyek ini membuktikan bahwa penggunaan alat sederhana seperti Excel dan VBA, jika dirancang dengan tepat, dapat menghasilkan solusi praktis yang langsung digunakan oleh tim dan berkontribusi terhadap peningkatan efektivitas kerja di lingkungan IT Security.

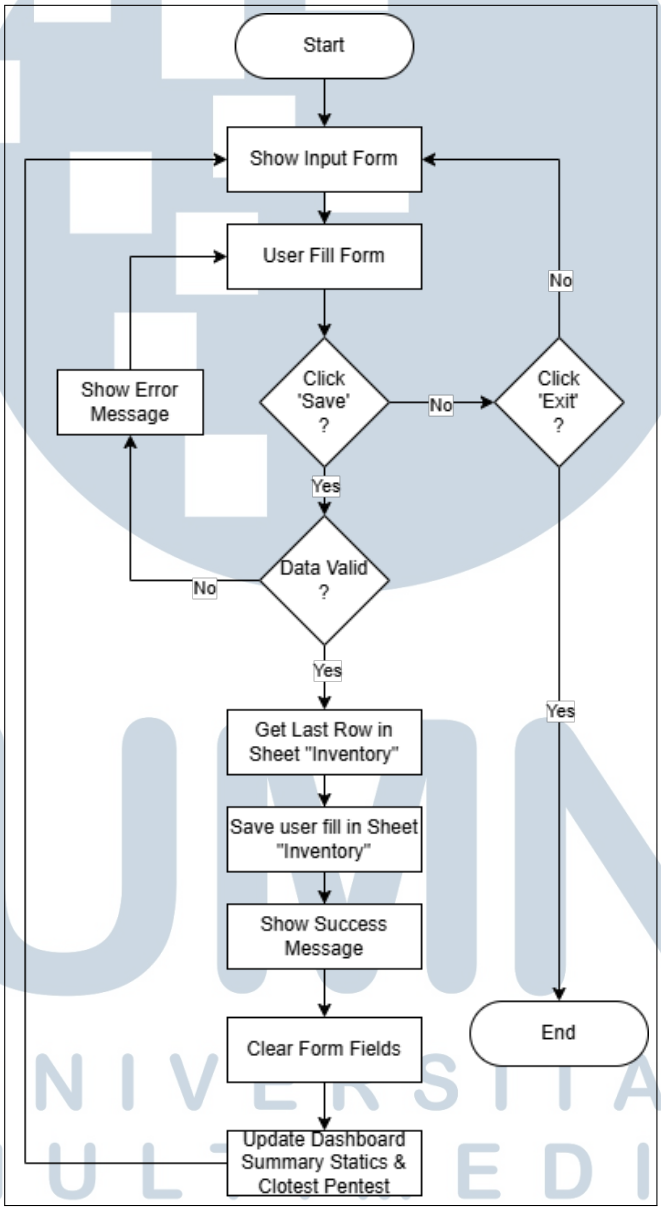
3.5.1 Dashboard Inventory Pentest

Pada Gambar 3.1 adalah gambaran alur proses yang ada pada halaman *home*(tampilan awal) *dashboard*. Di halaman ini pengguna dapat memilih fitur yang telah di sediakan, fitur berdasarkan status yaitu *open, closed, total* dan *exclude*. Fitur untuk *action* seperti *input, update dan read*. Fitur terakhir merupakan kategori berdasarkan *category* seperti *web, API, mobile android, mobile IOS* dan *all*.



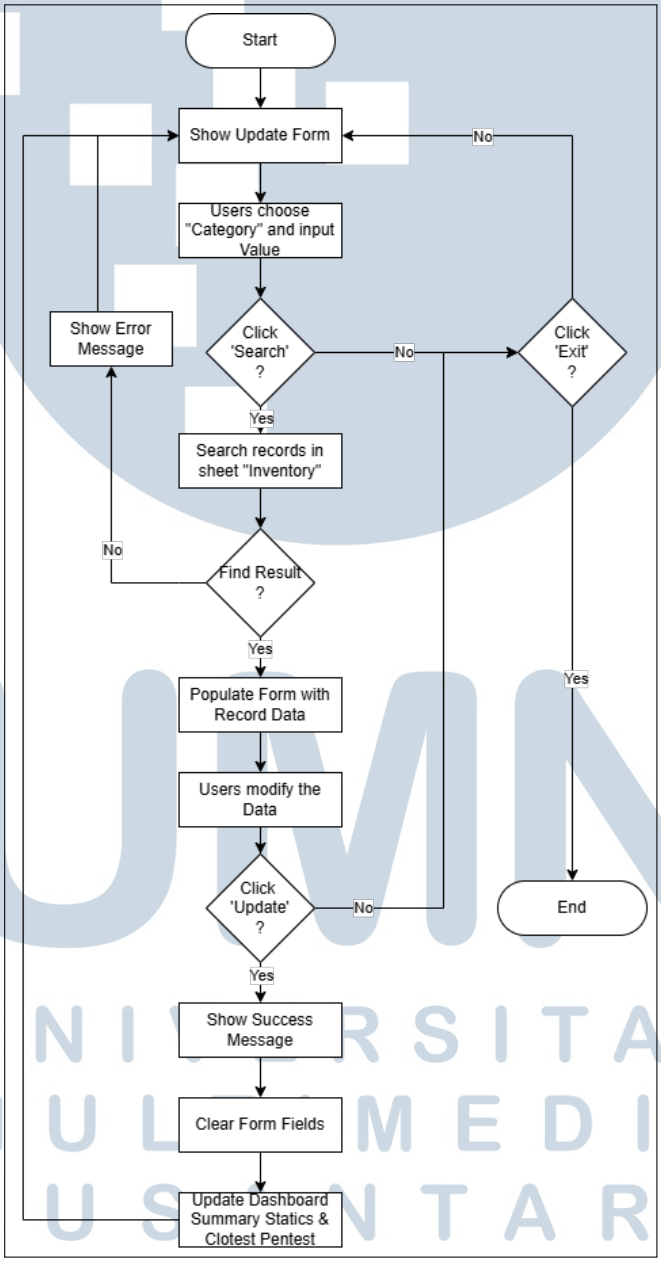
Gambar 3.1. Flowchart home base dashboard inventory penetration

Bagian Gambar 3.2 merupakan *flow process user* mengisi informasi *request* ataupun temuan yang di temukan lewat *form user input* dan nanti akan dilakukan pengujian oleh *pentester* sesuai informasi yang dimasukkan ke dalam *form input*, data baru akan disimpan pada *sheet* ”Inventory” nantinya akan mempengaruhi *summary statistic*, grafik dan informasi *pentest* terdekat di *dashboard*.



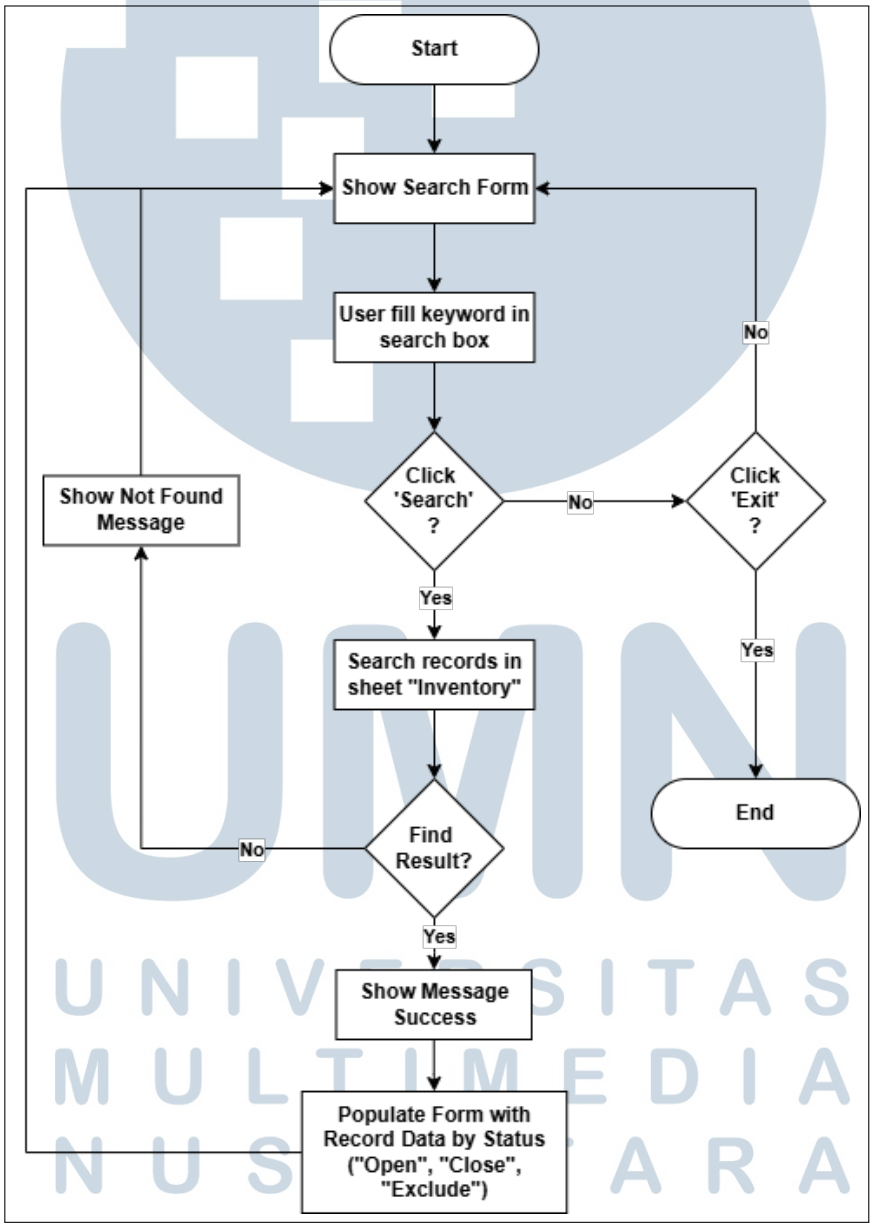
Gambar 3.2. Flowchart input form

Gambar 3.3 menunjukkan *flow* proses informasi atau data yang di *update* oleh *user*. Pengguna akan memilih kategori yang diubah dan masukan *value*. Jika data ditemukan, sistem akan memberikan notifikasi sukses dan *form* akan berubah menjadi informasi sesuai *value* yang dimasukkan. Perubahan yang dilakukan nantinya akan mempengaruhi tampilan *dashboard* bagian *summary statistic, graph* dan *closest pentest*



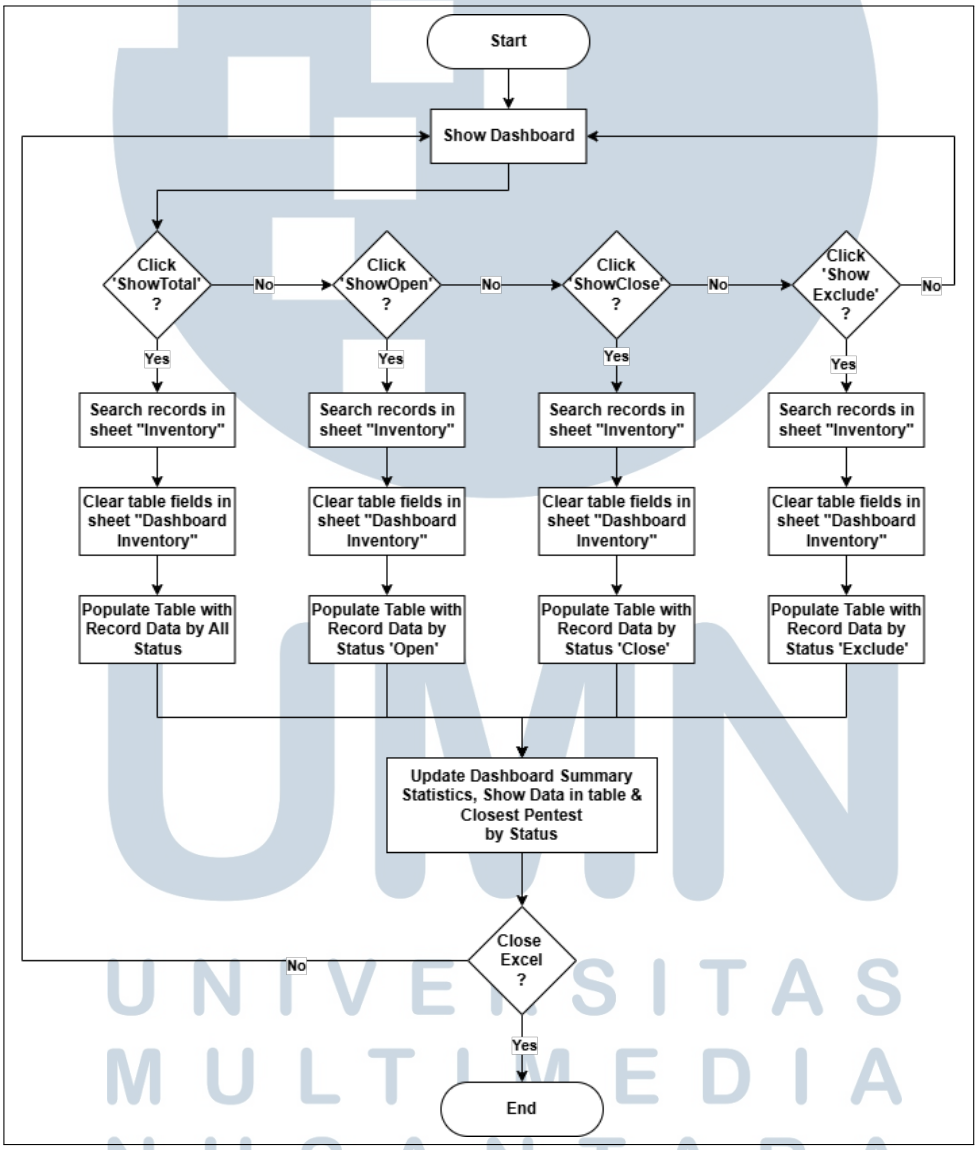
Gambar 3.3. Flowchart update form

Gambar 3.4 menampilkan alur proses pencarian data inventori oleh *user*. Sistem menampilkan *form* pencarian dan *user* memasukkan *keyword* sesuai data yang di cari. Setelah klik *search*, sistem mencari data di*sheet* ”Inventory”. Jika tidak di temukan, muncul pesan *Not Found* dan *user* kembali ke *form* pencarian. Jika di temukan, sistem menampilkan pesan sukses dan mengisi *form* dengan data sesuai status *(Open, Close, atau Exclude)*. Fitur ini memudahkan *user* mencari informasi permintaan pentest tanpa melihat seluruh *database*.



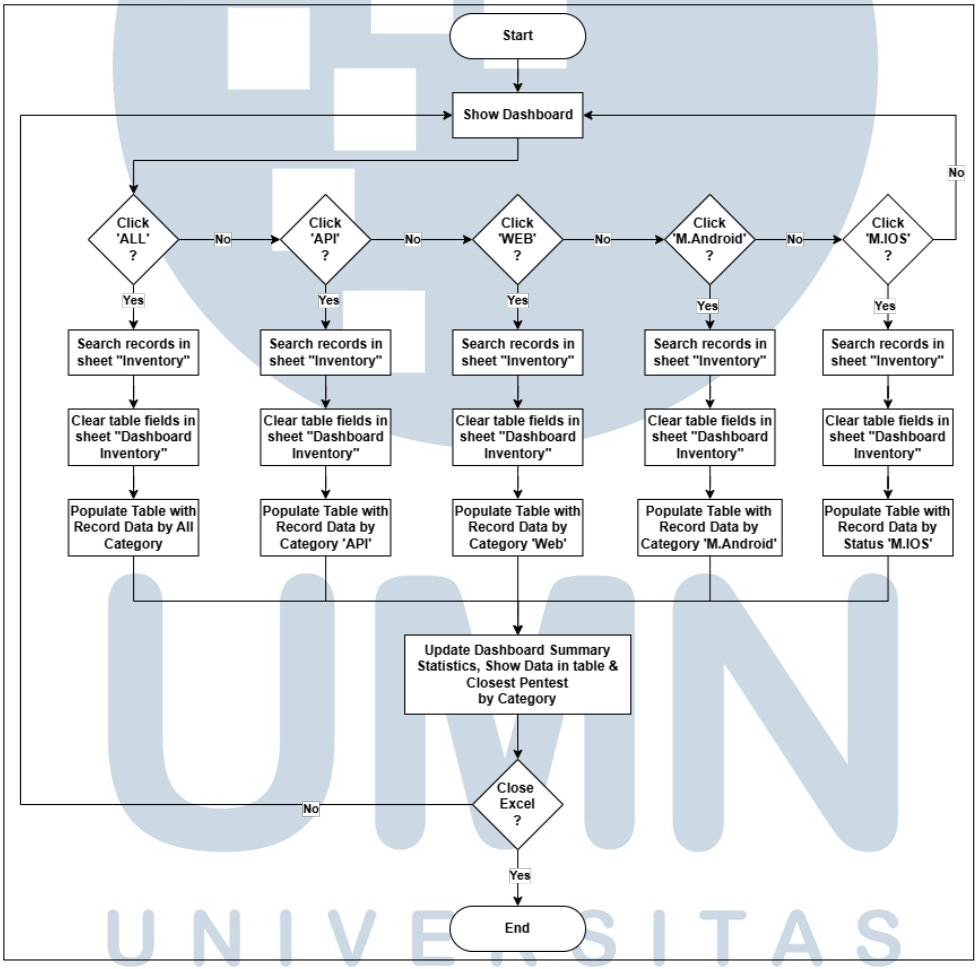
Gambar 3.4. Flowchart search form

Gambar 3.5 menunjukkan alur proses tampilan *dashboard* berdasarkan status permintaan *pentesting* yang dipilih. *Dashboard* menampilkan data request *pentesting* berdasarkan kategori diantaranya *ShowTotal*, *ShowClose*, *ShowOpen*, dan *ShowExclude*. Saat kategori dipilih, sistem mencari data di *sheet* "Inventory", menghapus data lama pada tabel *Dashboard Inventory*, lalu mengisi tabel dengan data hasil filter. *Summary statistic*, grafik, dan informasi jadwal *pentest* terdekat akan otomatis berubah sesuai kategori yang dipilih



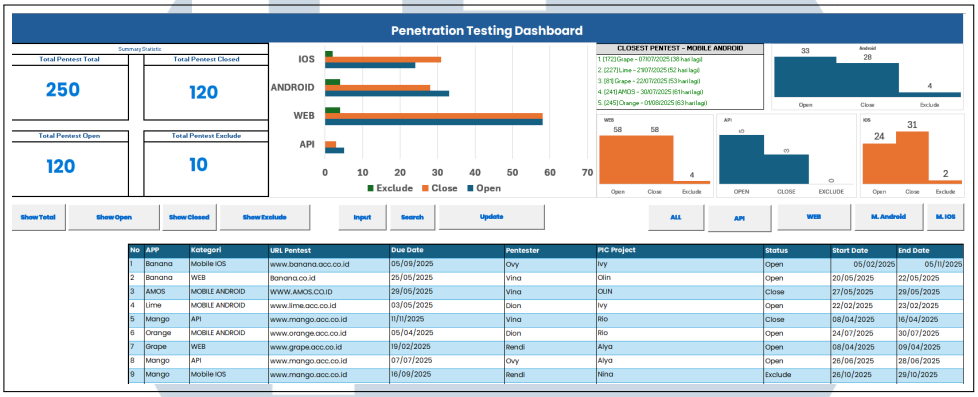
Gambar 3.5. Flowchart update dashboard by status

Gambar 3.6 menampilkan alur proses *dashboard* inventori berdasarkan kategori yang dipilih *user*. Terdapat lima kategori filter di antaranya, ALL, API, WEB, M.Android, dan M.IOS. Nantinya sistem memfilter data di *sheet* "Inventory", membersihkan data lama di tabel *sheet* "Dashboard Inventory", lalu menampilkan data sesuai kategori. Hasil *filtering* ini menampilkan data spesifik per *platform*, memudahkan *monitoring* inventori. Selanjutnya, sistem memperbarui *summary statistics*, tabel data, *closest pentest*, dan visualisasi *dashboard* sesuai filter yang dipilih.



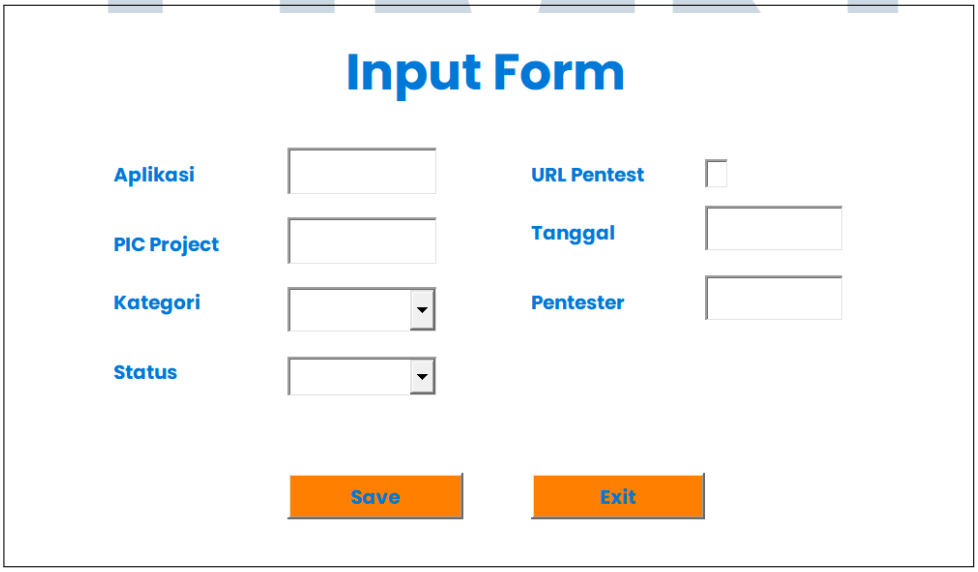
Gambar 3.6. Flowchart update dashboard by category

Gambar 3.7 menunjukkan *interface* utama sistem *dashboard penetration testing* yang menampilkan ringkasan statistik, visualisasi grafik berdasarkan platform (*iOS, Android, Web, API*), tombol filter berdasarkan status dan kategori, serta tabel yang menampilkan informasi data seperti nama, kategori, URL, *due date, pentester, PIC project*, dan status.



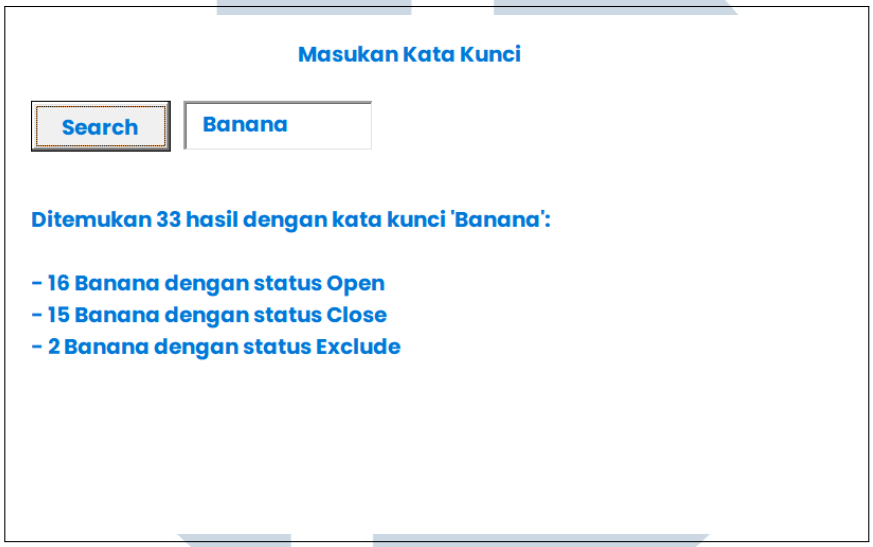
Gambar 3.7. Tampilan dashboard utama

Gambar 3.8 menampilkan *form input* untuk menambah data *penetration testing* baru. *Form* ini berisi *field* untuk mengisi nama aplikasi, URL, *PIC Project*, tanggal, kategori, status, *pentester, start date,*dan *end date*. Kategori dan status permintaan dipilih melalui *dropdown*, serta tersedia tombol *save* untuk menyimpan permintaan dan *exit* untuk membatalkan *input*.



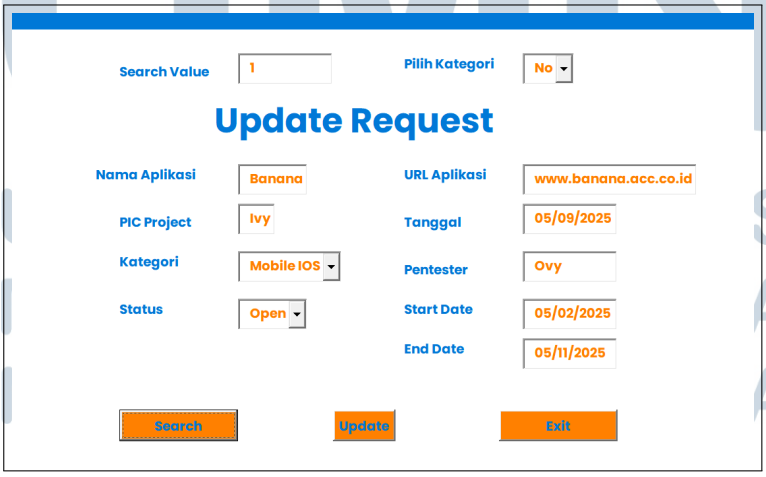
Gambar 3.8. Tampilan input form

Gambar 3.9 menampilkan formulir pencarian untuk *user* memasukkan kata kunci dan mencari data permintaan yang diperlukan. Sistem akan menampilkan data yang relevan dengan kata kunci. Fitur ini memudahkan *user* melakukan pencarian spesifik pada *database* inventori.



Gambar 3.9. Tampilan search form

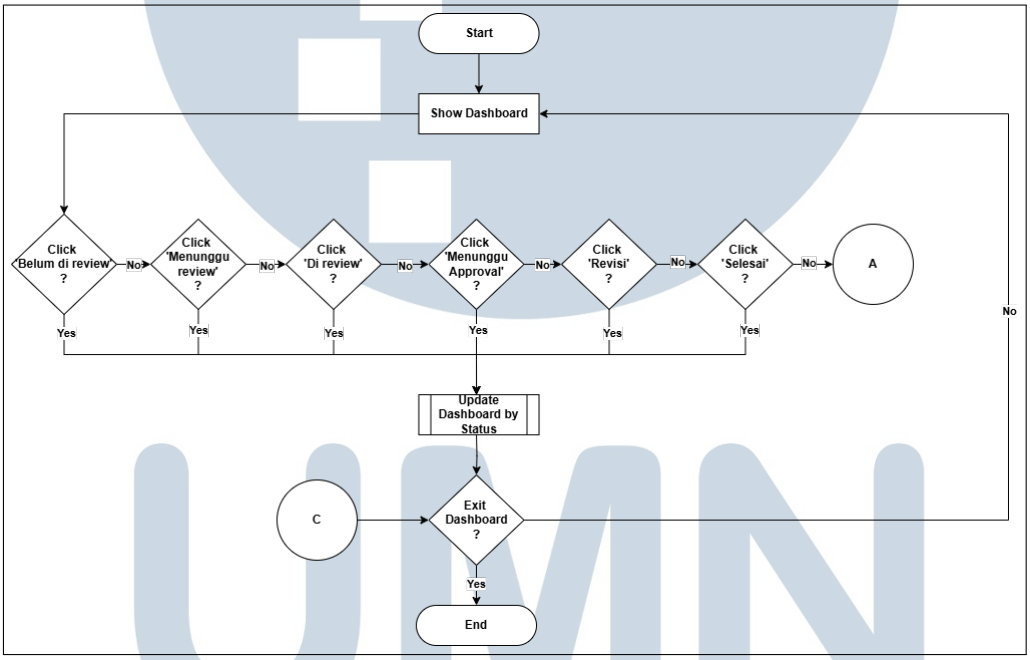
Gambar 3.10 menampilkan *form update* data yang otomatis terisi berdasarkan hasil pencarian. *Form* berisi data aplikasi beserta detail informasi yang dapat di *edit*, seperti nama aplikasi, URL, *PIC Project*, kategori, status , *pentester*, dan tanggal terkait. Tersedia tombol *search, update*, dan *exit* untuk pencarian, pembaruan, dan keluar.



Gambar 3.10. Tampilan update form

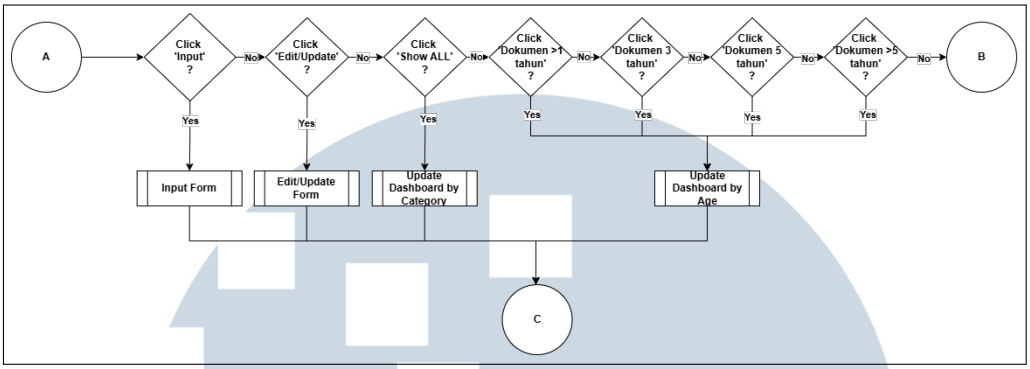
3.5.2 Dashboard Document Monitoring

Gambar 3.11 menggambarkan alur proses *update dashboard* berdasarkan status dokumen. Terdapat beberapa tombol status seperti*belum di-review*, *menunggu review*, di-*review*, *menunggu approval*, *revisi*, hingga *selesai*. Salah satu tombol status dokumen yang ditekan akan melakukan *update dashboard*, menampilkan tabel, visualisasi grafik, serta informasi kategori dokumen sesuai dengan status yang ditekan. Jika tidak ditekan, proses akan diteruskan ke proses *A*. Pada tahap ini, terdapat proses *C* yang merupakan kelanjutan dari proses *A* dan *B*. Proses kemudian diarahkan pada keputusan apakah pengguna ingin keluar dari *dashboard*. Jika pengguna memilih untuk keluar, maka proses akan dihentikan.



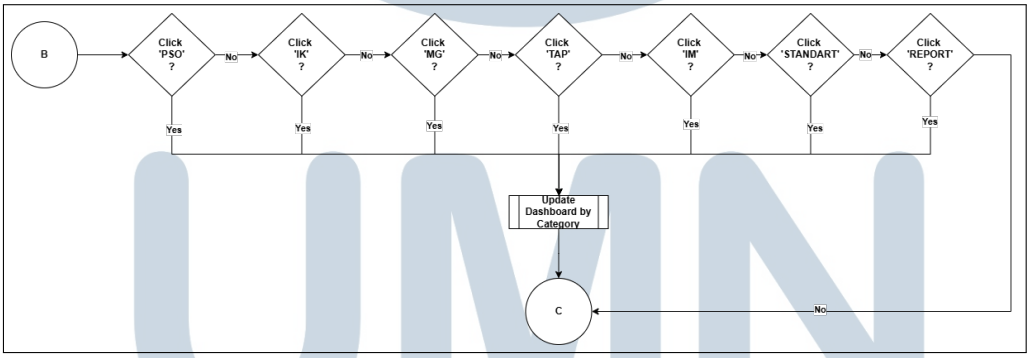
Gambar 3.11. Flowchart filter dashboard berdasarkan status dokumen

Gambar 3.12 menggambarkan alur proses *update dashboard* yang dimulai dari titik A atau melanjutkan alur proses sebelumnya. Jika pengguna tidak menekan tombol filter berdasarkan status dokumen, maka penggunakan dapat menekan tombol *form* seperti input atau *update*. Selain itu pengguna dapat menekan tombol filter dokumen berdasarkan tahun umur dokumen seperti satu tahun, tiga tahun dan lima tahun. Dari tombol yang di tekan, proses akan di lanjutkan dengan proses lain seperti membuka halaman formulir atau *update dashboard* berdasarkan tombol yang ditekan dan setelah itu proses akan kembali ke proses C. Namun jika tidak ada tombol yang di tekan, akan di lanjutkan ke proses B.



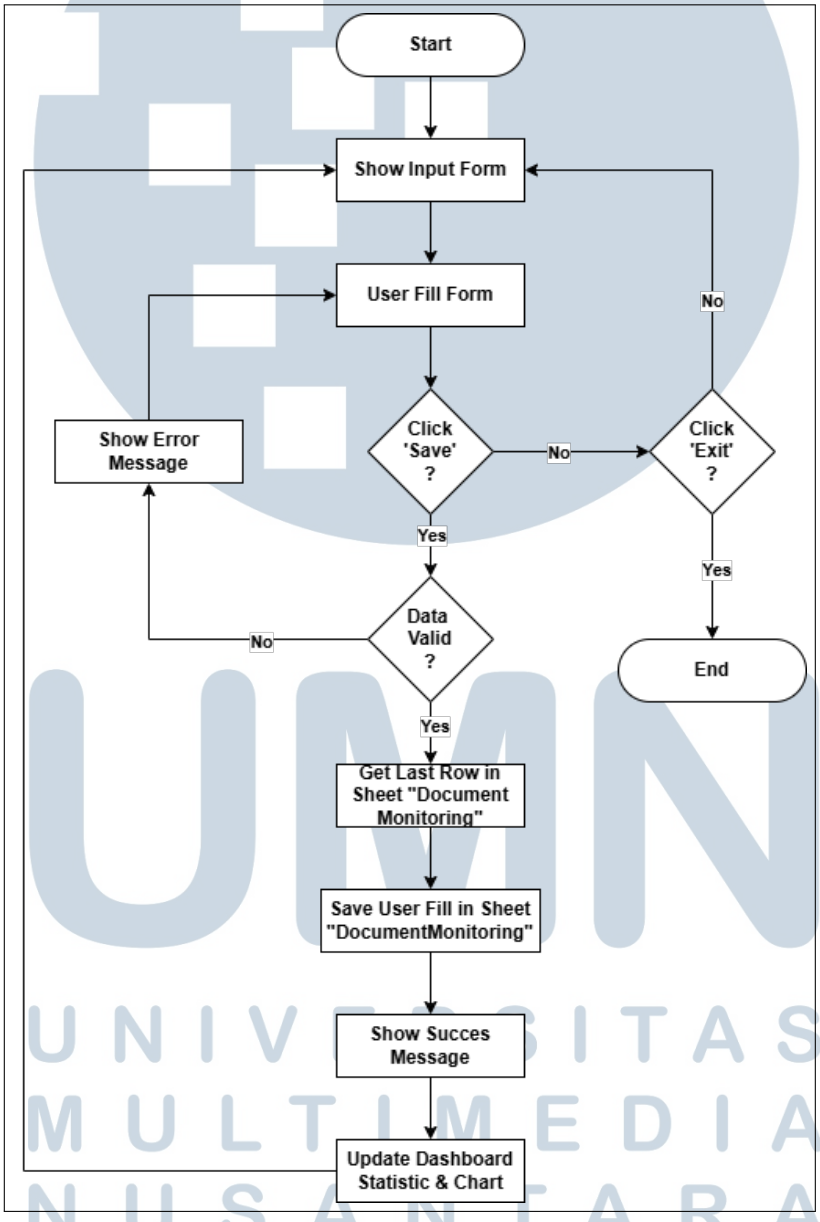
Gambar 3.12. Flowchart filter dashboard berdasarkan tipe form dan umur dokumen

Gambar 3.13 menampilkan alur proses *update dashboard* yang dimulai dari titik B atau melanjutkan alur proses sebelumnya. Pengguna dapat melakukan filter *dashboard* berdasarkan kategori dokumen seperti PSO, IK, Manual Guide (MG), TAP (Surat Ketetapan), IM (Internal Momerandum), standar, dan report. Dari tombol yang ditekan, nantinya akan dilanjutkan dengan *update dashboard*, tabel, dan visual grafik. Setelah *dashboard* di-*update* itu proses akan kembali ke proses C.



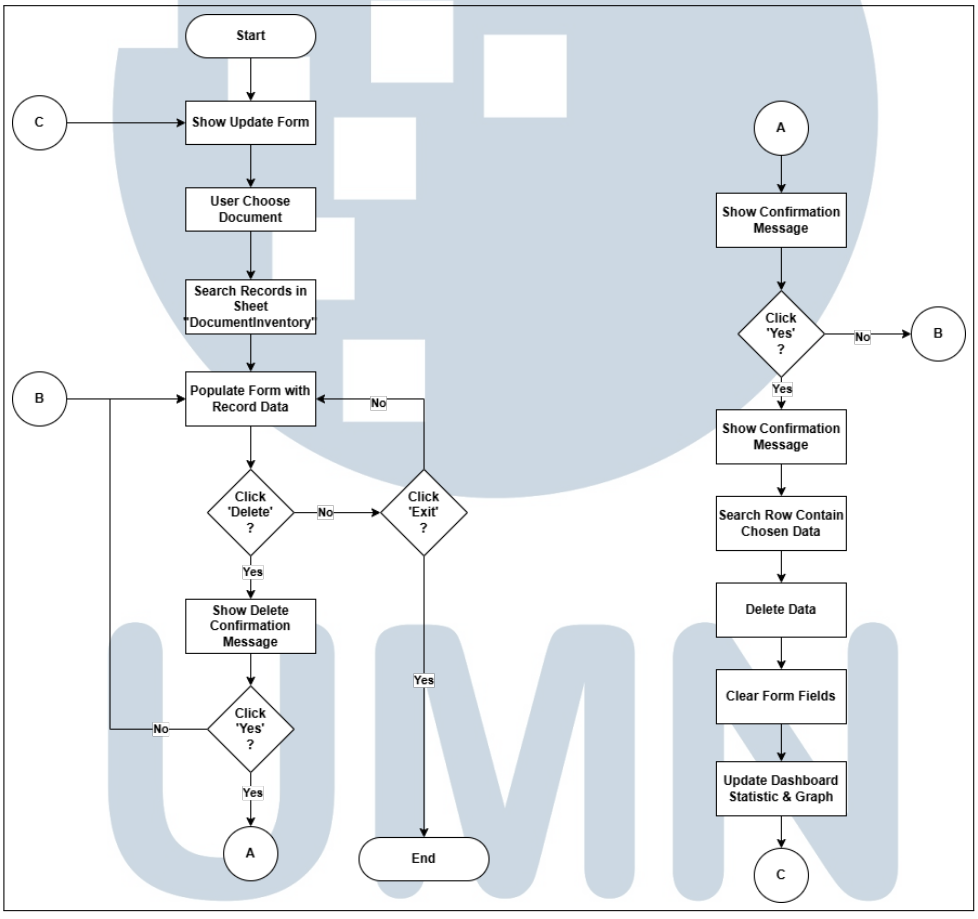
Gambar 3.13. Flowchart filter dashboard berdasarkan kategori dokumen

Gambar 3.14 menggambarkan alur penambahan data dokumen baru melalui *form input*. *User* mengisi data lalu memilih *save* atau *Exit*. Jika *Save*, sistem memvalidasi data. Jika valid, data di simpan di baris terakhir *sheet "DocumentMonitoring"*, pesan sukses di tampilkan, dan *dashboard* beserta *chart* di perbarui. Jika tidak valid, sistem tampilkan pesan *error* dan kembali ke *form input*.



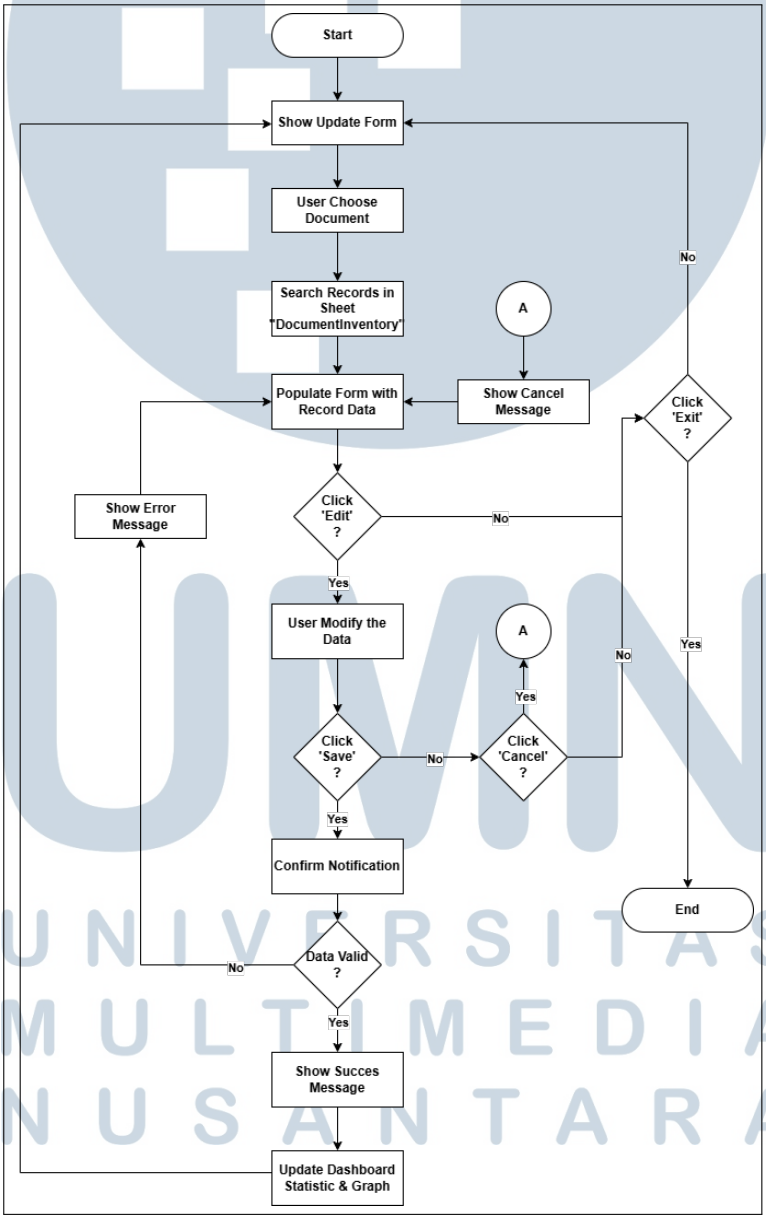
Gambar 3.14. Flowchart input data document

Gambar 3.15 menunjukkan alur khusus penghapusan data dari *form update*. Setelah *form* menampilkan data, *user* dapat memilih opsi *delete*. Sistem akan menampilkan pesan konfirmasi dengan opsi *yes* atau *no*. Jika *user* memilih *yes*, sistem mencari baris data terkait di-*sheet DocumentInventory*, menghapus data, membersihkan *field form*, memperbarui statistik *dashboard* dan grafik, lalu kembali ke halaman utama. Jika *No*, sistem membatalkan proses penghapusan dan tetap berada di *form update*.



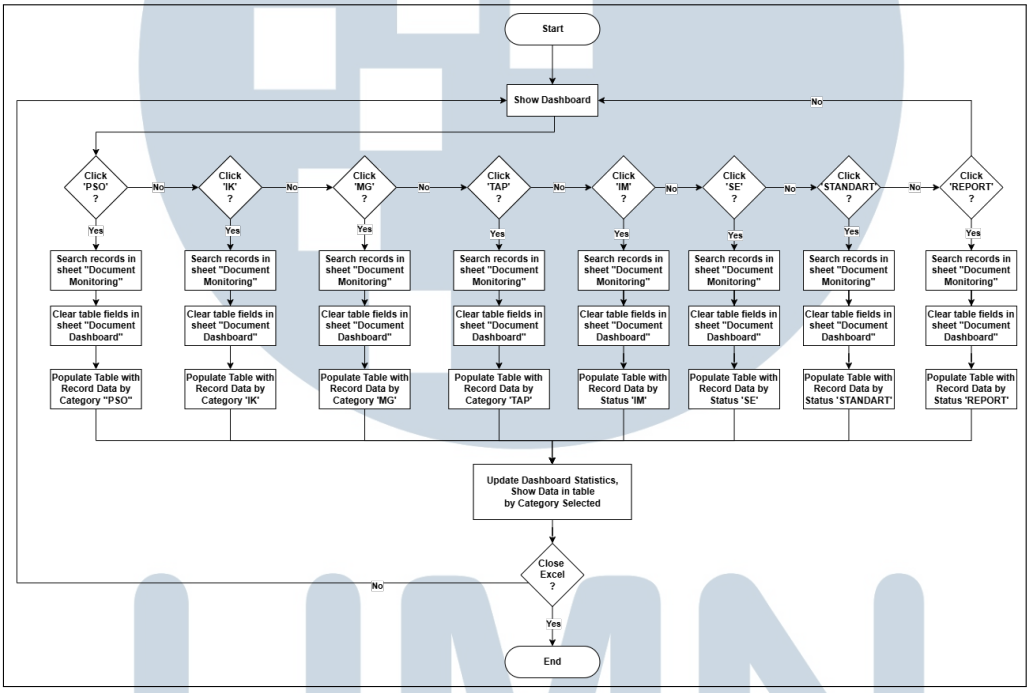
Gambar 3.15. Flowchart delete data document

Gambar 3.16 menggambarkan alur kerja sistem untuk mengedit data dokumen. Proses di mulai dengan menampilkan *form update*, lalu *user* memilih dokumen yang ingin di ubah. Sistem mencari data di-*sheet DocumentInventory* dan menampilkan isian data *existing* di-*form*. *User* dapat melakukan perubahan data, lalu klik *Update* untuk menyimpan. Sistem melakukan validasi data. Jika valid, sistem menyimpan perubahan, menampilkan pesan sukses, memperbarui statistik *dashboard*, lalu kembali ke halaman utama. Jika data tidak valid, sistem menampilkan pesan *error* dan mengarahkan kembali ke *form*.



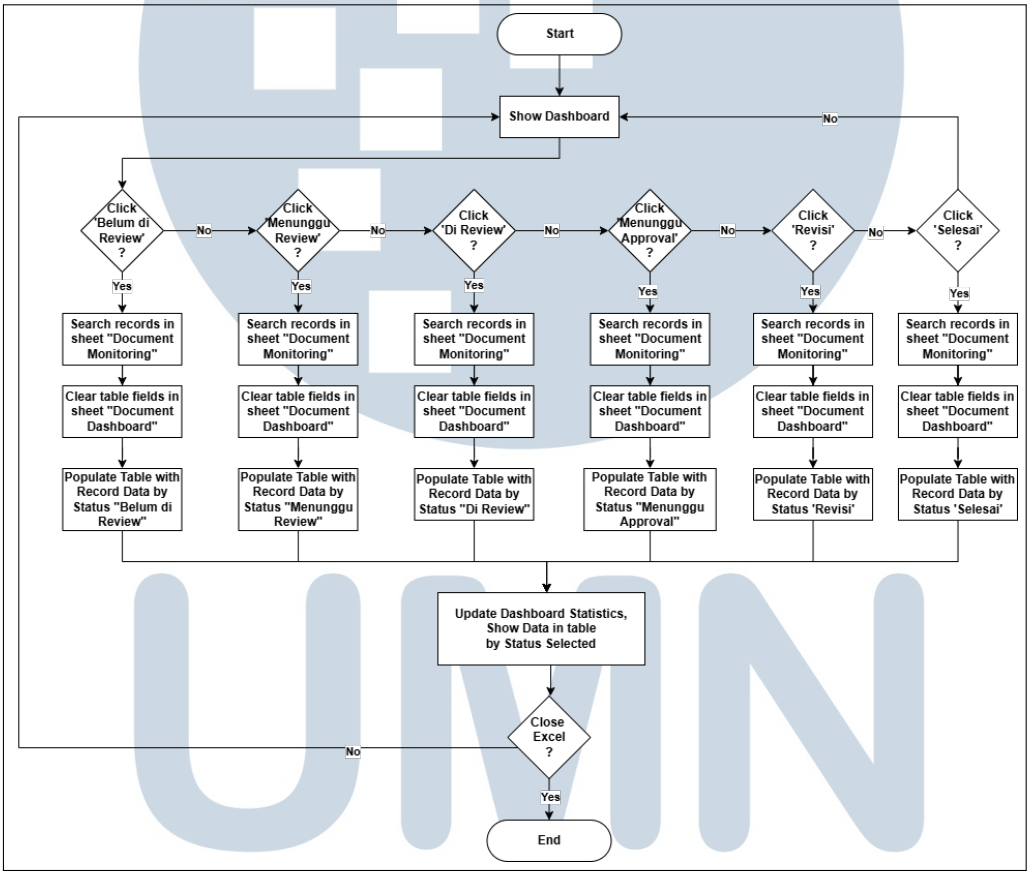
Gambar 3.16. Flowchart update data document

Gambar 3.17 menggambarkan alur *dashboard* yang di mulai dengan menampilkan halaman utama. Pengguna memilih kategori *filter* melalui *decision point* seperti PSO, IK, MG, TAP, IM, SE, STANDAR, dan REPORT, dengan opsi *Yes* atau *No*. Jika *Yes*, sistem mencari data di *sheet Document Monitoring*, membersihkan tabel di *Document Dashboard*, lalu mengisi tabel dengan data hasil *filter* sesuai kategori. Setelah proses selesai, sistem memperbarui statistik *dashboard* dan menampilkan data sesuai *filter* yang dipilih. Pengguna dapat memilih *Close Excel* atau melanjutkan proses lain sebelum keluar dari sistem.



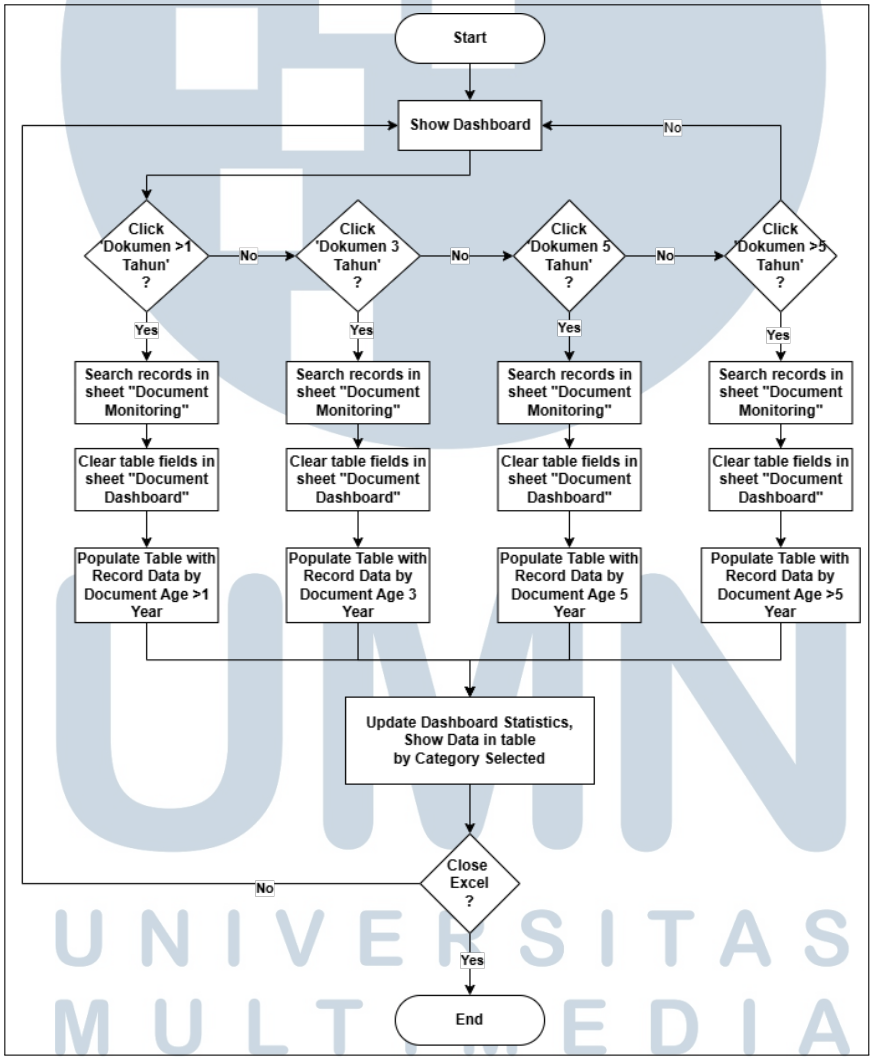
Gambar 3.17. Flowchart proses filtering dan tampilan dashboard berdasarkan kategori dokumen

Gambar 3.18 menampilkan alur kerja *filtering dashboard* berdasarkan kategori dokumen. Proses di mulai dengan *dashboard* utama, di mana *user* dapat memilih status seperti belum di-*review*, menunggu *review*, di-*review*, menunggu *approval*, revisi, dan selesai melalui *decision point yes/no*. Jika *yes*, sistem mencari data di-*sheet "DocumentMonitoring"*, membersihkan tabel di-*"DocumentDashboard*, lalu mengisi tabel dengan data hasil *filter*. Setelah *filtering* selesai, statistik *dashboard* di*update* dan hasil di tampilkan sesuai status terpilih. *user* dapat menutup *Excel* atau melanjutkan proses lain.



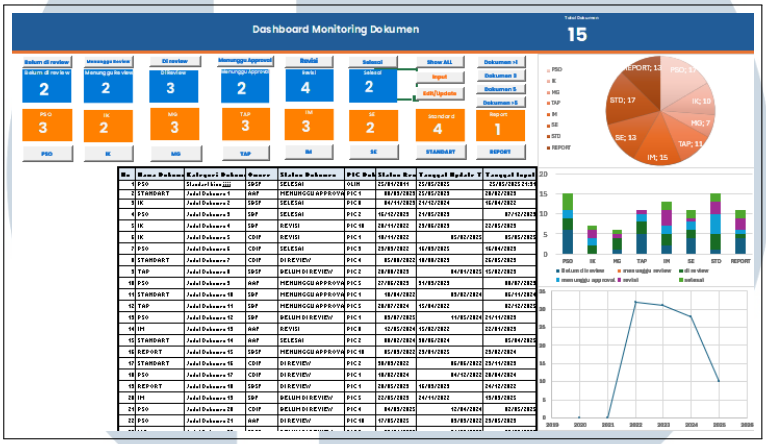
Gambar 3.18. Flowchart proses filtering dan tampilan dashboard berdasarkan status dokumen

Gambar 3.19 menggambarkan alur *filtering* data dokumen berdasarkan usia dokumen. Proses di mulai dengan menampilkan *dashboard* utama, lalu pengguna memilih kategori usia dokumen (1 Tahun, 3 Tahun, 5 Tahun, atau 5 Tahun). Jika dipilih, sistem mencari data di *sheet "DocumentMonitoring"*, membersihkan tabel lama di *Document Dashboard*, dan mengisi tabel dengan data sesuai kategori usia yang dipilih. Setelah itu, sistem memperbarui statistik *dashboard* dan menampilkan data yang telah di*filter*. Proses di akhiri dengan opsi menutup *Excel* atau melanjutkan.



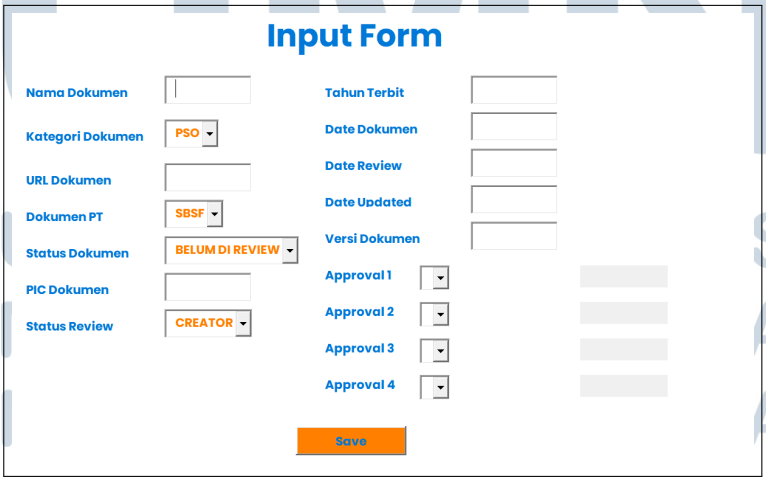
Gambar 3.19. Flowchart proses filtering dan tampilan dashboard berdasarkan tahun dokumen

Gambar 3.20 menampilkan *dashboard document monitoring* yang menampilkan informasi total dokumen, kategori dokumen, status dokumen, dan *approval* secara visual. *Dashboard* ini menyajikan data dalam bentuk tabel, diagram *pie*, diagram batang, dan grafik tren tahunan. Tersedia juga tombol *filter* berdasarkan kategori dokumen, dan status dokumen untuk memudahkan pencarian data tertentu.



Gambar 3.20. Tampilan dashboard document monitoring

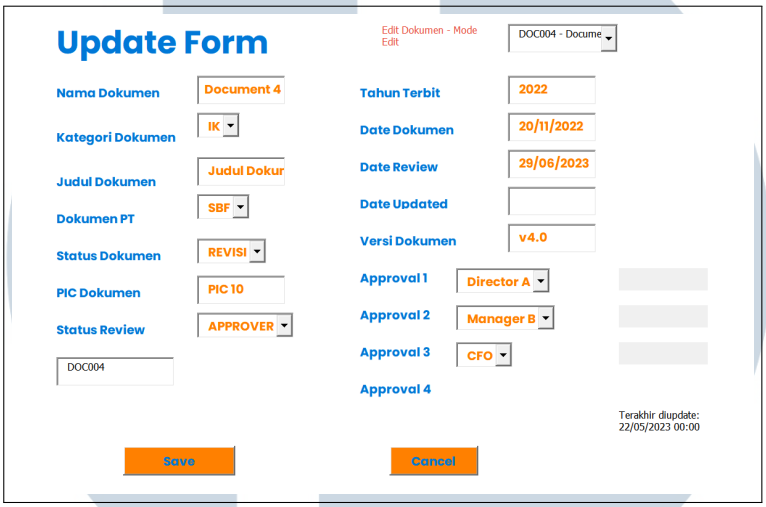
Gambar 3.21 Menjelaskan *form input* untuk menambahkan data dokumen baru ke dalam *Sheet "Inventory"*. Pengguna mengisi data seperti nama dokumen, kategori, URL, PT, status, PIC, tahun terbit, tanggal dokumen, tanggal *review*, versi dokumen, status *review*, dan *approval*. Setelah data lengkap, pengguna dapat menekan tombol *Save* untuk menyimpan data baru ke dalam sistem.



Gambar 3.21. Tampilan input form document monitoring

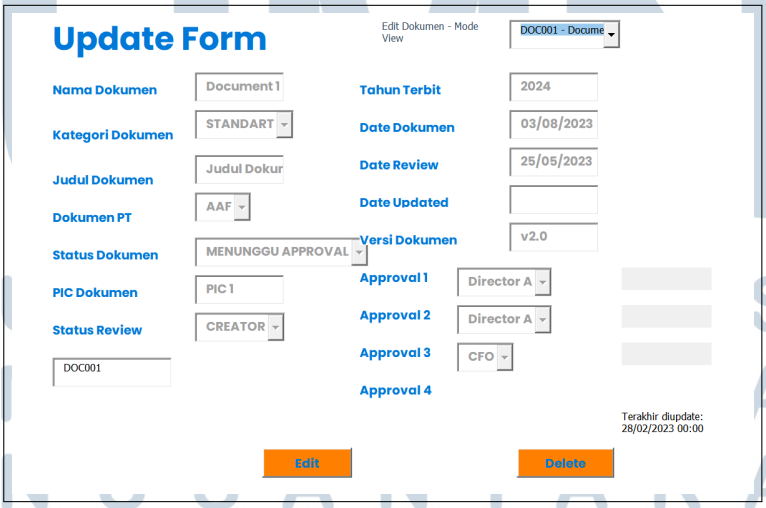
Gambar 3.22 menjelaskan tampilan *form update* yang berfungsi untuk

mengubah data dokumen yang telah ada di-*sheet "Inventory"*. Informasi dokumen yang dapat diubah seperti nama dokumen, kategori, *URL*, PT, status, PIC, tahun terbit, tanggal dokumen, tanggal *review*, versi dokumen, status *review*, dan *approval*.



Gambar 3.22. Tampilan update form document monitoring

Gambar 3.23 menampilkan tampilan *delete form*. Pengguna akan memilih dokumen yang akan di*delete*, kemudian sistem akan menampilkan informasi detail dokumen seperti kode dan nama. Ketika tombol *delete* diklik, nantinya akan ada *notification* untuk memastikan data dokumen yang dihapus sesuai.



Gambar 3.23. Tampilan delete form document monitoring

3.6 Kendala dan Solusi yang Ditemukan

Selama magang berjalan, terdapat kendala yang ditemukan sewaktu membuat sistem *monitoring* berbasis *Excel* dan *Visual Basic Application* dokumen di antaranya:

1. Sulit melakukan filter data dan menghitung jumlah data dokumen berdasarkan status dan kategori secara bersamaan menggunakan *Excel* biasa.
2. Data dokumen yang baru di-*input* pengguna tidak terhitung secara otomatis di-*dashboard document monitoring*.
3. Filter dokumen berdasarkan usia dokumen sejak *update* sulit untuk dibaca karena tidak ada format khusus untuk mengelompokkan umur.
4. Tabel di-*dashboard monitoring document* tidak otomatis berubah ketika tombol bagian status atau kategori dokumen di klik.

Solusi yang dapat diterapkan dari kendala yang ada dengan beberapa cara di antaranya:

1. Menambahkan tombol di-*sheet dashboard* yang terintegrasi antara *sheet database* dan *dashboard* dengan VBA (*Visual Basic Application*) serta logika untuk memunculkan hasil filter berdasarkan status atau kategori secara bersamaan berupa jumlah di label dan tabel untuk informasi detail dokumen.
2. Menggunakan VBA untuk menambahkan fungsi untuk melakukan perhitungan otomatis dengan cara men-*update* jumlah dokumen di label dan *chart* berdasarkan tombol yang di klik, seperti input, kategori ataupun status dokumen.
3. Menambahkan tombol yang terintegrasi antara *sheet database* dan *dashboard* berbasis VBA dengan menambahkan logika fungsi kategori usia dokumen untuk melakukan filter data dokumen seperti, dokumen berusia satu tahun, dokumen usia tiga tahun, dan usia lima tahun sejak di-*update*.
4. Menambah fungsi menggunakan VBA untuk membersihkan tabel di*sheet dashboard* setiap tombol filter dokumen (status, kategori dan usia) diklik dan memunculkan hasil filter di tabel yang sudah dibersihkan.