

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan pesat dalam teknologi dan persaingan global yang semakin ketat menuntut industri manufaktur, khususnya di sektor otomotif, untuk terus berinovasi demi mencapai efisiensi operasional yang maksimal. Efisiensi ini tidak hanya diukur dari kecepatan produksi, tetapi juga dari kemampuan untuk mengelola aset, meminimalkan waktu henti *downtime*, dan memastikan setiap komponen yang digunakan berada dalam kondisi optimal. Salah satu aset paling krusial namun seringkali kompleks untuk dikelola dalam sebuah lini produksi adalah peralatan kerja atau *tools*, yang secara langsung memengaruhi kualitas produk akhir dan kelancaran proses manufaktur.

PT Suzuki Indomobil merupakan salah satu perusahaan manufaktur otomotif terkemuka di Indonesia yang mengoperasikan lini produksi berskala besar dan berteknologi tinggi. Dalam operasionalnya, PT Suzuki Indomobil memanfaatkan ribuan jenis *tool* spesifik, mulai dari cetakan *moulding*, mata bor, hingga alat-alat presisi lainnya. Setiap *tool* memiliki karakteristik unik, termasuk masa pakai yang ditentukan oleh jumlah penggunaan *shot count* dan umur teknis *lifetime* dalam satuan jam kerja. Keberlangsungan produksi sangat bergantung pada ketersediaan dan kondisi prima dari *tools* tersebut.

Sebelum adanya sistem terpusat, proses manajemen dan pelacakan untuk aset-aset vital ini dihadapkan pada sejumlah tantangan. Pencatatan yang dilakukan secara *semi-manual*, seringkali menggunakan *spreadsheet* terpisah untuk setiap departemen, terbukti kurang efektif dalam lingkungan produksi yang dinamis. Beberapa kendala signifikan yang teridentifikasi antara lain adalah kesulitan dalam melacak kondisi *tool* secara *real-time*, yang berisiko menyebabkan penggunaan melebihi batas amannya. Selain itu, manajemen perbaikan yang kurang terstruktur seringkali menyebabkan ketidakpastian mengenai ketersediaan *tool*. Hal ini diperparah dengan rendahnya

visibilitas terhadap stok suku cadang yang terintegrasi, sehingga proses perbaikan menjadi lebih lambat. Pada akhirnya, proses rekapitulasi data untuk laporan menjadi sangat memakan waktu dan rentan terhadap kesalahan manusia *human error* karena data yang tersebar di berbagai dokumen.

Menyadari adanya kebutuhan untuk mengatasi tantangan-tantangan tersebut dan sebagai bagian dari upaya digitalisasi proses bisnis di lingkungan perusahaan, maka dalam pelaksanaan kerja praktek ini, penulis ditugaskan untuk merancang dan membangun sebuah solusi digital yang komprehensif. Solusi tersebut diwujudkan dalam bentuk sistem aplikasi berbasis *web* dengan nama *Tool Tracking System*.

Aplikasi ini dirancang secara spesifik untuk menjadi pusat kendali tunggal dalam manajemen *tools* di PT Suzuki. Dengan fitur-fitur seperti pelacakan *shot count* dan jam pakai otomatis, manajemen alur kerja perbaikan yang terstruktur, pemantauan stok suku cadang, serta dasbor pelaporan yang interaktif, sistem ini bertujuan untuk mentransformasi proses yang sebelumnya *reaktif* menjadi lebih *proaktif*. Diharapkan, implementasi *Tool Tracking System* ini tidak hanya akan meningkatkan efisiensi operasional, tetapi juga menyediakan data yang akurat dan dapat diandalkan untuk mendukung pengambilan keputusan strategis di tingkat manajerial.

1.2 Tujuan dan Manfaat

1.2.1 Tujuan Umum Magang

Tujuan dari pelaksanaan magang di PT Suzuki Indomobil adalah untuk memenuhi salah satu syarat kelulusan program studi Mesin Otomotif di Politeknik Negeri Jember, serta untuk mengaplikasikan ilmu pengetahuan teoritis yang telah diperoleh selama perkuliahan ke dalam lingkungan kerja yang nyata. Tujuan umum dari pelaksanaan magang ini adalah sebagai berikut:

1. Memahami dan merasakan secara langsung atmosfer serta budaya kerja profesional di sebuah perusahaan berskala besar seperti PT Suzuki.

2. Membangun kemampuan adaptasi, komunikasi, dan kerja sama tim dalam lingkungan kerja yang sesungguhnya.
3. Memperoleh wawasan dan pengalaman praktis mengenai penerapan teknologi informasi dalam mendukung proses bisnis di industri manufaktur otomotif.
4. Mampu membandingkan dan menghubungkan antara teori yang didapat di perkuliahan dengan implementasi praktis di lapangan.

1.2.2 Tujuan Khusus Magang Mahasiswa

Tujuan khusus yang ingin dicapai melalui proyek pembangunan aplikasi *Tool Tracking System* ini adalah:

1. Mampu menganalisis kebutuhan sistem dan mengidentifikasi permasalahan yang ada dalam proses manajemen *tool* di PT Suzuki.
2. Mampu merancang sebuah sistem aplikasi berbasis *web* yang fungsional, mulai dari desain *database*, alur kerja pengguna, hingga antarmuka (*UI/UX*).
3. Berhasil mengimplementasikan fitur-fitur utama sesuai kebutuhan perusahaan, seperti pelacakan *shot count* dan *lifetime*, manajemen perbaikan, pemantauan stok, dan sistem pelaporan.
4. Menguasai dan menerapkan teknologi pengembangan *web* modern, khususnya menggunakan *framework Django Python* untuk *backend* dan *Bootstrap* untuk *frontend*.
5. Menghasilkan sebuah produk aplikasi yang fungsional dan siap untuk diimplementasikan, yang dapat memberikan solusi nyata terhadap masalah yang telah diidentifikasi.

1.2.3 Manfaat Magang Mahasiswa

Pelaksanaan magang ini diharapkan dapat memberikan manfaat:

1. Mendapatkan pengalaman kerja nyata yang sangat berharga dalam pengembangan perangkat lunak dari awal hingga akhir (end-to-end).
2. Meningkatkan kemampuan teknis (*hard skills*) dalam bahasa pemrograman *Python, framework Django*, manajemen database, dan teknologi *web* lainnya.
3. Mengembangkan kemampuan non-teknis (*soft skills*) seperti manajemen waktu, pemecahan masalah (*problem-solving*), dan komunikasi profesional dengan rekan kerja dan pembimbing.
4. Memperoleh bahan dan data yang sangat relevan untuk menyusun laporan magang yang berkualitas.
5. Membangun jaringan profesional dan mendapatkan gambaran yang lebih jelas mengenai karir di bidang mesin otomotif setelah lulus.

1.3 Lokasi dan Waktu

1.3.1 Lokasi Magang



Gambar 1. 1 Denah Perusahaan

Tempat pelaksanaan magang di PT. Suzuki Indomobil Tambun Jl. Diponegoro Km. 38,2 Kel. Jatimulya Kec. Tambun Selatan, Kab. Bekasi, Jawa Barat, 17510.

1.3.2 Waktu Magang

Kegiatan magang di PT. Suzuki Indomobil Tambun dimulai pada tanggal 18 Agustus 2025 s/d 30 Januari 2026. Selama periode tersebut, penulis melaksanakan seluruh kegiatan dengan disiplin, mematuhi semua peraturan yang berlaku, serta hadir dan bekerja sesuai dengan jadwal jam kerja yang telah ditetapkan oleh perusahaan.

1.4 Metode pelaksanaan

Metode pelaksanaan yang digunakan dalam pengerjaan proyek aplikasi *Tool Tracking System* ini adalah pendekatan pengembangan perangkat lunak secara bertahap *phased development*. Setiap tahapan diselesaikan secara berurutan untuk memastikan fondasi yang kokoh sebelum melangkah ke tahap berikutnya. Proses ini mencakup analisis kebutuhan, perancangan sistem, implementasi *coding*, pengujian, hingga *deployment*.

1.4.1 Analisis dan Perancangan

Tahap awal ini berfokus pada pemahaman masalah dan perancangan solusi.

1. Wawancara dan Pengumpulan Kebutuhan: Melakukan diskusi mendalam dengan pembimbing lapangan untuk mengidentifikasi masalah-masalah yang ada dalam sistem manajemen *tool* manual, serta merumuskan fitur-fitur utama yang dibutuhkan oleh aplikasi.
2. Pemilihan Teknologi: Berdasarkan kebutuhan akan aplikasi *web* yang dinamis dan memiliki *database*, dipilihlah tumpukan teknologi berikut:
 - a) *Backend: Python* dengan *framework Django*.
 - b) *Frontend: HTML, CSS, dan JavaScript* dengan *library Bootstrap 5*.
 - c) *Database: SQLite* untuk pengembangan lokal dan *PostgreSQL / MySQL* untuk produksi.
 - d) *Version Control: Git* dan *GitHub*.

3. Perancangan *Database*: Mendesain struktur *database* dengan membuat model-model di *Django*, seperti *Tool*, *Proses*, *UserProfile*, *Riwayat Perbaikan*, *Suku Cadang*, dan lain-lain untuk merepresentasikan semua entitas data yang dibutuhkan.

1.4.2 Implementasi Backend

Tahap ini adalah pembangunan mesin dan logika inti dari aplikasi.

1. Inisialisasi Proyek: Menyiapkan lingkungan pengembangan lokal, membuat *virtual environment*, dan menginisialisasi proyek *Django* baru.
2. Pembuatan Model dan Migrasi: Menulis kode untuk semua model yang telah dirancang di *models.py* dan menerapkan strukturnya ke dalam *database* menggunakan perintah *makemigrations* dan *migrate*.
3. Pengembangan Logika *Views*: Membangun semua fungsi logika di dalam *views.py* untuk setiap fitur, termasuk proses CRUD (*Create*, *Read*, *Update*, *Delete*) untuk data *tool*, alur kerja pengiriman *tool* ke *lab maintenance*, logika kalkulasi *shot count* dan *lifetime*, serta sistem hak akses.
4. Pembuatan *Form*: Merancang semua *formulir input* yang dibutuhkan pengguna di *forms.py*, seperti *form* untuk menambah *tool* baru, *form registrasi*, dan *form input produksi*.
5. Konfigurasi URL: Mengatur semua alamat URL di *urls.py* untuk setiap halaman dan fungsi agar aplikasi dapat diakses dengan benar.

1.4.3 Implementasi Frontend

Setelah mesin jadi, tahap ini berfokus pada pembuatan antarmuka yang dilihat oleh pengguna.

1. Pembuatan *Template* Dasar: Merancang satu *template* utama (*base.html*) yang berisi struktur dasar aplikasi, termasuk *sidebar* navigasi dan *layout* utama, untuk memastikan tampilan yang konsisten di semua halaman.
2. Integrasi *Bootstrap*: Menerapkan *library Bootstrap 5* untuk menata komponen-komponen seperti tabel, kartu (*card*), tombol, dan *formulir* agar memiliki tampilan yang *modern* dan *responsif*.
3. Pengembangan *Template* Halaman: Membuat *file-file HTML* individual untuk setiap halaman (daftar *tool*, laporan, manajemen stok, dll.) dan menghubungkannya dengan data yang dikirim dari backend.
4. Penambahan Interaktivitas: Menggunakan *JavaScript* untuk menambahkan fitur-fitur dinamis seperti jam digital yang berdetak, grafik donat yang interaktif di halaman laporan, dan *pop-up* (modal) untuk menambah data atau melihat riwayat.

1.4.4 Pengujian dan Debugging

Pengujian dilakukan secara iteratif di setiap akhir implementasi fitur untuk memastikan fungsionalitas berjalan sesuai harapan.

- a) Pengujian Fungsional: Mencoba semua alur kerja, seperti mendaftarkan pengguna baru, menambah *tool*, mengirimnya ke lab, menyelesaikan perbaikan, dan mengunduh laporan, untuk memastikan tidak ada *error*.
- b) *Debugging*: Mengidentifikasi dan memperbaiki berbagai jenis *error* yang muncul selama pengembangan, seperti *NameError* (karena lupa *import*), *FieldError* (karena model dan *form* tidak sinkron), *AttributeError* (karena salah memanggil nama properti), hingga *error Improperly Configured* saat proses *deployment*.

1.4.5 Deployment

Tahap terakhir adalah mempublikasikan aplikasi dari lingkungan lokal ke *server online* agar dapat diakses oleh pengguna.

1. Persiapan Kode: Membuat *file requirements.txt* untuk mencatat semua *library* yang dibutuhkan dan *.gitignore* untuk mengabaikan *file* yang tidak perlu diunggah.
2. Unggah ke *GitHub*: Menggunakan *Git* untuk mengunggah versi *final* dari kode aplikasi ke repositori di *GitHub*.
3. Konfigurasi *Server*: Melakukan pendaftaran dan konfigurasi di *platform hosting PythonAnywhere*, termasuk membuat *web app* baru, menarik kode dari *GitHub*, dan menyiapkan *virtual environment* di *server*.
4. Konfigurasi Produksi: Menyesuaikan *file settings.py* untuk mode produksi (*DEBUG = False*), mengatur *ALLOWED_HOSTS*, dan mengkonfigurasi *file WSGI* serta *static files* agar aplikasi dapat berjalan dengan benar di lingkungan *online*.