

**RANCANG BANGUN APLIKASI MONITORING PRODUKSI HARIAN
MANDOR UNIT TEBENAN PT. PERKEBUNAN NUSANTARA VII
BERBASIS ANDROID**

(Skripsi)

Oleh

MUHAMMAD IQBAL SYAHBANA



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

**RANCANG BANGUN APLIKASI MONITORING PRODUKSI
HARIAN MANDOR UNIT TEBENAN PT. PERKEBUNAN
NUSANTARA VII BERBASIS ANDROID**

Oleh
MUHAMMAD IQBAL SYAHBANA

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada
**Program Studi Teknik Informatika
Fakultas Teknik**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

ABSTRAK

RANCANG BANGUN APLIKASI MONITORING PRODUKSI HARIAN MANDOR UNIT TEBENAN PT. PERKEBUNAN NUSANTARA VII BERBASIS ANDROID

Oleh

MUHAMMAD IQBAL SYAHBANA

Pencatatan hasil produksi tanaman karet sangat penting untuk memantau kinerja produksi dan membuat keputusan bisnis yang tepat. Pada penelitian ini dikembangkan aplikasi monitoring produksi harian dengan adanya otentikasi pengguna sebagai keamanan data pada aplikasi. Penelitian ini bertujuan untuk membangun sebuah sistem monitoring dan pencatatan produksi harian tanaman karet berbasis *android* untuk mandor Unit Tebenan PT. Perkebunan Nusantara. Aplikasi ini menyediakan fitur pencatatan hasil produksi serta monitoring berupa grafik hasil karet setiap bulannya, selain itu juga terdapat informasi mengenai petunjuk pencatatan karet yang dilakukan oleh mandor. Metode yang digunakan dalam mengembangkan aplikasi pencatatan produksi harian ini adalah *agile kanban*. Pengembangan aplikasi ini dilakukan selama 45 hari dengan total 21 *backlog*. Proses pengujian aplikasi dilakukan dengan metode *blackbox testing* dengan total 12 skenario pengujian. Hasil pengujian *black box* menyatakan bahwa seluruh skenario menghasilkan keluaran sesuai dengan yang diharapkan. Sehingga, didapatkan kesimpulan bahwa aplikasi pencatatan dan monitoring produksi harian karet sudah berjalan sesuai kebutuhan fungsional dan non-fungsional aplikasi.

Kata Kunci: *Android, Flutter, Monitoring produksi karet, Metode Kanban, Blackbox Testing*

ABSTRACT

DEVELOPMENT OF ANDROID-BASED APPLICATION FOR DAILY PRODUCTION MONITORING BY THE SUPERVISOR OF TEBENAN UNIT AT PT. PERKEBUNAN NUSANTARA VII

By

MUHAMMAD IQBAL SYAHBANA

Recording the results of rubber plant production is very important to monitor production performance and make appropriate business decisions. In this study, an application for daily production monitoring was developed with user authentication as a data security measure. The aim of this research is to build an Android-based system for monitoring and recording daily rubber plant production for the supervisor of Tebenan Unit at PT. Perkebunan Nusantara. The application provides features for recording production results and monitoring rubber plant yields in the form of monthly graphs, as well as information on rubber recording instructions for the supervisor. The agile Kanban method was used in developing this daily production recording application. The development of this application was carried out for 45 days with a total of 21 backlogs. The application testing process was conducted using black-box testing with a total of 12 testing scenarios. The black-box testing results indicate that all scenarios produce the expected output. Therefore, it can be concluded that the rubber daily production recording and monitoring application meets both functional and non-functional requirements.

Keyword: *Android, Flutter, Rubber Production Monitoring, Kanban Method, Blackbox Testing*

Judul Skripsi : **RANCANG BANGUN APLIKASI MONITORING
PRODUKSI HARIAN MANDOR UNIT TEBENAN
PT. PERKEBUNAN NUSANTARA VII
BERBASIS ANDROID**

Nama Mahasiswa : **Muhammad Iqbal Syahbana**

Nomor Pokok Mahasiswa : 1915061009

Program Studi : Teknik Informatika

Fakultas : Teknik



Pembimbing Utama

Pembimbing Pendamping

Ir. Meizano Ardhi Muhammad, S.T., M.T.
NIP 19810528 201212 1 001

Mahendra Pratama, S.T., M.Eng.
NIP 19911215 201903 1 013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika


Herlinawati, S.T., M.T.
NIP 19710314 199903 2 001

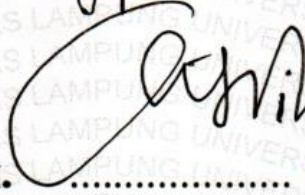
Mona Arif Muda, S.T., M.T.
NIP 19711112 200003 1 002

MENGESAHKAN

1. Tim Penguji

Ketua : **Ir. Melzano Ardhi Muhammad, S.T., M.T.** 

Sekretaris : **Mahendra Pratama, S.T., M.Eng.** 

Penguji : **Ir. Gigih Forda Nama, S.T., M.T.I., IPM.** 

2. Dekan Fakultas Teknik


Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. 
NIP 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **25 Mei 2023**

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini , menyatakan bahwa skripsi saya dengan judul “Rancang Bangun Aplikasi Monitoring Produksi Harian Mandor Unit Tebenan PT. Perkebunan Nusantara VII Berbasis Android” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 12 Juni 2023

Pembuat pernyataan,



Muhammad Iqbal Syahbana

NPM 1915061009

RIWAYAT HIDUP



Penulis dilahirkan di Kotaagung, pada tanggal 01 Juni 2001. Penulis merupakan anak pertama dari pasangan Bapak Emsah, S.Pd. dan Ibu Banatul Khoiriyah, S.Ag.

Penulis menyelesaikan pendidikannya di SD Negeri 4 Kuripan pada tahun 2013, SMP Negeri 1 Kotaagung pada tahun 2016, dan SMA Negeri 1 Kotaagung pada tahun 2019. Pada tahun 2019 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SNMPTN.

Selama menjalani proses perkuliahan secara aktif, penulis mengikuti Himpunan Mahasiswa Teknik Elektro (HIMATRO) sebagai anggota Divisi Minat dan Bakat pada Periode 2020 dan menjadi anggota Divisi Minat dan Bakat pada Periode 2021. Pada organisasi luar kampus, penulis juga mengikuti komunitas Gerakan Digital Ekosistem Nusantara (GRADIEN) yang sekaligus menjadi wadah penulis dalam mengembangkan ilmu di bidang teknologi informasi. Selain proses perkuliahan, penulis juga pernah melaksanakan studi independent kampus merdeka di Dicoding Indonesia dan magang di PT. Perkebunan Nusantara VII yang bertempat di Bandar Lampung sebagai Software Developer dan PT. United Tractors Tbk. yang bertempat di Jakarta Timur sebagai IT developer.

Prestasi yang pernah dicapai penulis antara lain adalah sebagai siswa terbaik pada studi independent di Dicoding Indonesia dengan nilai yang memuaskan dan mendapat beasiswa di Dicoding Indonesia pada tahun 2021. Lolos kegiatan wirausaha mahasiswa ke tahap nasional dengan membawa project *Molondry App* (Aplikasi Laundry). Keahlian penulis adalah pengembangan *mobile apps dan website* khususnya *cross platform Flutter dan react Js*.

MOTTO

“Menetapkan tujuan adalah langkah awal dalam mengubah yang tidak terlihat menjadi yang terlihat.”

(Penulis)

“Allah akan meninggikan orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(Q.S. Al-Mujadalah : 11)

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu.”

(Ali Bin Abi Thalib)

“Jangan pergi mengikuti kemana jalan akan berujung. Buat jalanmu sendiri dan tinggalkanlah jejak.”

(Ralph Waldo Emerson)

“Kamu tidak harus menjadi hebat untuk memulai, tetapi kamu harus mulai untuk menjadi hebat.”

(Zig Ziglar)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Sujud syukur kupersembahkan kepada Allah سُبْحَانَهُ وَتَعَالَى , Tuhan Yang Maha Esa dan Maha Besar. Berkat limpahan rahmat-Mu saya bisa menjadi pribadi yang bertaqwa, beriman, dan berilmu. Semoga dengan keberhasilan yang telah dicapai ini saya dapat menuju masa depan yang lebih baik dan dapat menggapai cita-cita serta selalu berada di jalan-Mu.

KUPERSEMBAHKAN SKRIPSI INI
TERUNTUK:

“Ibunda Banatul Khoiriyah dan Ayahanda Emsah atas dukungan dan kasih sayang yang diberikan mulai dari saya ada di dunia ini sampai saya sudah besar seperti sekarang ini. Terima kasih kepada Ibu dan Ayah atas doa yang tak henti-hentinya dipanjatkan serta pengorbanan yang tak terhitung nilainya. Semoga dengan ilmu dan cita-cita yang saya dapatkan kelak akan menjadi amal jariyah bagi Ibu dan Ayah”

“Terima Kasih untuk adik-adikku Fathan Rizqi Syahbana dan Ghania Mahira Syahbana yang selalu bermain denganku saat dirumah. Semoga kalian kelak menjadi pribadi yang lebih sukses.”

“Diriku sendiri. Maaf untuk malam-malam panjang dengan mata yang sulit tertidur, kepala yang sakit, dan lelah pikiran. Terima Kasih telah berjuang dan bekerjasama selama ini. Kamu hebat.”

“Terima Kasih kepada teman-teman Teknik Informatika 2019 yang telah menemani dan membantu saya selama perkuliahan di kampus tercinta Universitas Lampung. Semoga kelak kita semua akan menjadi orang-orang yang sukses.”

“Terima Kasih kepada seseorang yang selalu memberikan semangat dan motivasi saat saya jatuh dan yang selalu mengingatkan saya untuk bersyukur saat saya bahagia. Semoga Allah selalu memberikan kita kebahagiaan di apapun jalan yang kita pilih kedepannya.”

SANWACANA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayat-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini dengan judul “Rancang Bangun Aplikasi Monitoring Produksi Harian Mandor Unit Tebenan PT. Perkebunan Nusantara VII Berbasis Android”.

Dalam pelaksanaan dan pembuatan skripsi ini penulis menerima dukungan baik secara moril maupun materil yang sangat berharga dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu, khususnya kepada:

1. Allah سُبْحَانَهُ وَ تَعَالَى yang senantiasa memberikan kemudahan dan kelancaran kepada penulis serta Rasulullah Muhammad صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ yang telah menjadi suri tauladan selama penelitian berlangsung.
2. Ibu dan Ayah serta keluarga penulis yang selalu memberikan motivasi dan dukungan kepada penulis.
3. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
4. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
5. Bapak Mona Arif Muda, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan telah membantu proses kelancaran pengerjaan penelitian;
6. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T., selaku Pembimbing Utama penelitian yang selalu meluangkan waktunya untuk memberikan bimbingan dan dukungan;

7. Bapak Mahendra Pratama, S.T., M.Eng., selaku Pembimbing Pendamping penelitian yang selalu memberikan motivasi dan memberikan bimbingan kepada penulis untuk menjadi lebih baik;
8. Bapak Ir. Gigih Forda Nama, S.T., M.T.I., IPM., selaku Pembimbing Akademik yang telah memberikan bimbingan selama perkuliahan disetiap semester dan selalu memberikan motivasi serta telah bersedia menjadi penguji dalam sidang skripsi.
9. Mbak Rika selaku Admin Program Studi Teknik Informatika yang telah banyak membantu penulis dalam segala urusan administrasi selama perkuliahan;
10. Seluruh dosen dan staf Jurusan Teknik Informatika Unila yang memberi masukan dan mempermudah proses pembuatan skripsi ini.
11. Teman-teman seperjuangan khususnya Rachel Adella Ferninda, teman-teman Grup Laut, dan Ben 9, yang telah banyak memberikan saran, dukungan, dan bantuannya.
12. Semua pihak yang turut serta dalam membantu menyelesaikan penelitian dan tidak mungkin penulis sebutkan satu persatu

Akhir kata, semoga skripsi ini dapat bermanfaat bagi pembacanya.

Bandar Lampung, 12 Juni 2023

Penulis,

Muhammad Iqbal Syahbana

DAFTAR ISI

| | Halaman |
|---|----------------|
| PERSEMBAHAN..... | iv |
| SANWACANA | iv |
| DAFTAR ISI | iv |
| DAFTAR GAMBAR..... | vii |
| DAFTAR TABEL | ix |
| | |
| I. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Tujuan Penelitian | 3 |
| 1.4 Manfaat Penelitian | 3 |
| 1.5 Batasan Masalah | 4 |
| 1.6 Sistematika Penulisan | 4 |
| | |
| II. TINJAUAN PUSTAKA | 6 |
| 2.1 Tanaman Karet..... | 6 |
| 2.2 PT. Perkebunan Nusantara VII | 7 |
| 2.3 Monitoring Hasil Produksi | 7 |
| 2.4 <i>Kanban Method</i> | 8 |
| 2.5 <i>Android</i> | 9 |
| 2.5.1 <i>Local Storage Android</i> | 9 |
| 2.6 <i>Application Programming Interface (API)</i> | 10 |
| 2.6.1 <i>Open Weather Map API</i> | 11 |
| 2.6.2 <i>Postman API</i> | 11 |
| 2.7 <i>Global Positioning System (GPS)</i> | 12 |
| 2.8 <i>Javascript Object Notation (JSON)</i> | 13 |
| 2.9 <i>Blackbox testing</i> | 14 |
| 2.10 <i>Computer Aided Software Engineering tools</i> | 17 |
| 2.10.1 <i>Visual Studio Code</i> | 17 |
| 2.10.2 <i>Flutter</i> | 18 |
| 2.10.3 <i>Git</i> | 19 |
| 2.10.4 <i>Github</i> | 20 |
| 2.10.5 <i>CRUD Booster</i> | 21 |
| 2.10.6 <i>Trello</i> | 22 |

| | | |
|-------------|--|-----------|
| 2.11 | Penelitian Terkait..... | 22 |
| 2.11.1 | <i>Development and Application of Condition Monitoring System for Plant Production</i> | 22 |
| 2.11.2 | <i>Design of the Information System for Kindergarten Learning Evaluation used Kanban Methodology</i> | 24 |
| 2.11.3 | Implementasi Metodologi Kanban Dalam Pembuatan Aplikasi <i>E-Commerce</i> Pertanian Dengan Pendekatan <i>Zachman Framework</i> | 26 |
| 2.12 | <i>State of The Art</i> | 27 |
| III. | METODE PENELITIAN | 29 |
| 3.1 | Waktu dan Tempat Penelitian..... | 29 |
| 3.2 | Jadwal Penelitian | 29 |
| 3.3 | Alat Dalam Penelitian | 30 |
| 3.3.1 | Alat Penelitian | 30 |
| 3.3.2 | Bahan Penelitian..... | 31 |
| 3.4 | Tahapan Penelitian..... | 31 |
| 3.4.1 | Target Pengguna..... | 32 |
| 3.4.2 | Tahap Penentuan <i>User story</i> | 32 |
| 3.4.3 | Tahap Perancangan..... | 33 |
| 3.4.4 | Tahap Pengembangan Sistem..... | 41 |
| 3.4.5 | Tahap <i>Testing</i> | 42 |
| 3.4.6 | Tahap Analisis | 42 |
| 3.4.7 | Tahap Pelaporan | 43 |
| IV. | HASIL DAN PEMBAHASAN..... | 44 |
| 4.1 | Hasil | 44 |
| 4.1.1 | <i>User story</i> | 44 |
| 4.1.2 | Perancangan..... | 45 |
| 4.1.3 | Pengembangan..... | 50 |
| 4.1.4 | Evaluasi | 63 |
| 4.2 | Pembahasan | 70 |
| 4.2.1 | Konfigurasi Sistem | 70 |
| 4.2.2 | Analisis Keterbatasan | 73 |
| 4.2.3 | Kegiatan Pengembangan | 73 |
| 4.2.4 | Capaian Penelitian | 92 |
| V. | SIMPULAN DAN SARAN | 94 |
| 5.1 | Simpulan | 94 |
| 5.2 | Saran | 95 |
| | DAFTAR PUSTAKA..... | 96 |
| | LAMPIRAN | 99 |

DAFTAR GAMBAR

| | Halaman |
|--|----------------|
| Gambar 2.1. Workflow kanban | 8 |
| Gambar 2.2. Android Version | 9 |
| Gambar 2.3. Workspace Postman | 12 |
| Gambar 2.4. Cara kerja Global Positioning System (GPS) | 13 |
| Gambar 2.5. Contoh Format JSON | 14 |
| Gambar 2.6. Workspace VS Code | 18 |
| Gambar 2.7. Flutter | 19 |
| Gambar 2.8. Git | 20 |
| Gambar 2.9. GitHub | 21 |
| Gambar 2.10. CRUD Booster | 21 |
| Gambar 2.11. Kanban menggunakan Trello | 22 |
| Gambar 2.12. <i>Kanban board</i> yang digunakan dalam penelitian..... | 24 |
| Gambar 2.13. <i>Kanban board</i> yang digunakan pada penelitian..... | 26 |
| Gambar 3.1. Tahapan pada metode Kanban | 31 |
| Gambar 3.2. Format <i>user story</i> yang digunakan dalam penelitian | 33 |
| Gambar 3.3. Task card | 34 |
| Gambar 3.4. Environment System | 35 |
| Gambar 3.5. <i>Splash Screen</i> | 36 |
| Gambar 3.6. <i>Login Screen</i> | 37 |
| Gambar 3.7. <i>Home Screen</i> | 38 |
| Gambar 3.8. <i>Produksi Harian Screen</i> | 39 |
| Gambar 3.9. <i>Performa Mandor Screen</i> | 40 |
| Gambar 3.10. <i>Performa Produktivitas Mandor Screen</i> | 41 |
| Gambar 4.1. Backlog pada perancangan aplikasi | 46 |

| | |
|--|----|
| Gambar 4.2. Rancangan Icon Aplikasi | 47 |
| Gambar 4.3. Rancangan Splash Screen Aplikasi | 47 |
| Gambar 4.4. Rancangan Laman Login Aplikasi | 48 |
| Gambar 4.5. Rancangan Laman Home Screen | 48 |
| Gambar 4.6. Rancangan Laman Produksi Harian | 49 |
| Gambar 4.7. Rancangan Laman Pencatatan Produksi Harian..... | 49 |
| Gambar 4.8. Laman grafik produksi mandor | 50 |
| Gambar 4.9. Tampilan laman Login aplikasi..... | 51 |
| Gambar 4.10. Tampilan kode program laman Login aplikasi | 52 |
| Gambar 4.11. Tampilan icon logout aplikasi | 53 |
| Gambar 4.12. Tampilan kode program proses logout aplikasi | 54 |
| Gambar 4.13. Tampilan laman produksi harian | 55 |
| Gambar 4.14. Tampilan kode program laman produksi harian | 56 |
| Gambar 4.15. Tampilan laman update produksi harian | 57 |
| Gambar 4.16. Tampilan kode program laman update produksi harian..... | 58 |
| Gambar 4.17. Tampilan laman monitoring produksi mandor..... | 59 |
| Gambar 4.18. Tampilan kode program monitoring produksi harian mandor | 59 |
| Gambar 4.19. Tampilan asterisk pada saat aktif | 60 |
| Gambar 4.20. Tampilan asterisk pada saat tidak aktif | 60 |
| Gambar 4.21. Tampilan kode program asterik pada password..... | 61 |
| Gambar 4.22. Dialog saat login berhasil..... | 62 |
| Gambar 4.23. Tampilan kode program dialog login | 62 |
| Gambar 4.24. Android versi 6..... | 69 |
| Gambar 4.25. Android versi 11 | 69 |
| Gambar 4.26. Membuat akun pada layanan Open Weather Map | 70 |
| Gambar 4.27. Kunci API pada Open Weather Map..... | 71 |
| Gambar 4.28. Konfigurasi Open Weather Map di Visual Studio Code..... | 72 |
| Gambar 4.29. Konfigurasi Manifest pada Visual Studio Code..... | 72 |
| Gambar 4.30. Daftar user story sistem..... | 74 |
| Gambar 4.31. Format taskcard yang digunakan pada kanban board | 75 |
| Gambar 4.32. Label yang digunakan dalam kanban board..... | 75 |
| Gambar 4.33. Breakdown user story pertama..... | 77 |

| | |
|---|----|
| Gambar 4.34. Breakdown user story kedua | 78 |
| Gambar 4.35. Breakdown user story ketiga | 79 |
| Gambar 4.36. Kanban board pada penelitian yang dilakukan | 80 |
| Gambar 4.37. Kanban board pada penelitian yang dilakukan | 81 |
| Gambar 4.38. Kanban board pada penelitian yang dilakukan | 82 |
| Gambar 4.39. Kanban board pada penelitian yang dilakukan | 83 |
| Gambar 4.40. Backlog perancangan antarmuka aplikasi | 84 |
| Gambar 4.41. Proses perancangan antarmuka | 85 |
| Gambar 4.42. Backlog inisiasi pyorek dan konfigurasi sistem | 86 |
| Gambar 4.43. Backlog pengembangan dari fitur otentikasi login | 87 |
| Gambar 4.44. Backlog pengembangan dari logout aplikasi | 88 |
| Gambar 4.45. Backlog pengembangan splash Screen aplikasi | 88 |
| Gambar 4.46. Backlog pengembangan dari fitur pencatatan produksi harian | 89 |
| Gambar 4.47. Backlog pengembangan dari fitur monitoring hasil produksi | 90 |

DAFTAR TABEL

| | Halaman |
|--|----------------|
| Tabel 3.1. Jadwal Penelitian..... | 29 |
| Tabel 3.2. Alat yang digunakan dalam penelitian..... | 30 |
| Tabel 3.3. Daftar user story..... | 33 |
| Tabel 4.1. Daftar user story pada system | 45 |
| Tabel 4.2. Hasil pengujian blackbox testing pada otentikasi login..... | 65 |
| Tabel 4.3. Hasil pengujian blackbox testingpada otentikasi logout..... | 66 |
| Tabel 4.4. Hasil pengujian blackbox testingpada pencatatan produksi harian | 67 |
| Tabel 4.5. Hasil pengujian blackbox testingpada fitur monitoring..... | 68 |
| Tabel 4.6. Rincian waktu pengembangan | 91 |
| Tabel 4.7. Rincian waktu pengujian..... | 92 |

I. PENDAHULUAN

1.1 Latar Belakang

Pencatatan produksi tanaman karet sangat penting untuk memantau kinerja produksi dan membuat keputusan bisnis yang tepat. Setiap hari, jumlah getah yang dipanen dari setiap pohon harus didokumentasikan secara detail. Ini membantu dalam mengidentifikasi tren produksi dan menentukan kebutuhan perbaikan. Selain itu, pencatatan produksi juga melibatkan pemantauan kondisi lingkungan seperti cuaca, suhu, dan curah hujan serta perawatan tanaman seperti pemupukan, penyiangan dan pemotongan. Data ini sangat berguna dalam memahami bagaimana faktor lingkungan dan perawatan mempengaruhi produksi tanaman.

PT Perkebunan Nusantara VII memanfaatkan dokumen manual sebagai media pencatatan produksi harian tanaman karet. Buku *log* atau jurnal yang berkualitas baik dapat membantu memudahkan catatan dan analisis produksi harian. Dalam buku *log* tercatat jumlah produksi yang diperoleh setiap hari, kondisi lingkungan seperti cuaca dan suhu, serta tindakan perawatan tanaman yang dilakukan. Kelebihan dari penggunaan dokumen manual adalah tidak tergantung pada jaringan internet. Informasi produksi, kondisi lingkungan, dan perawatan tanaman dicatat secara rinci dan dapat dianalisis untuk membantu pengambilan keputusan bisnis yang tepat. Dokumen manual juga membantu memonitor perkembangan produksi tanaman karet dari waktu ke waktu, sehingga perusahaan dapat membuat tindakan yang tepat untuk meningkatkan produksi. Terlepas dari dokumen manual, PTPN VII memiliki sistem yang terintegrasi yang dapat menyimpan data hasil keseluruhan produksi karet setiap unitnya, namun data tersebut didapatkan dari data hasil pencatatan manual oleh mandor [1].

Dokumen manual juga memiliki beberapa kekurangan. Dokumen manual rawan hilang atau rusak, sulit dicari dan dibandingkan dengan data lainnya. Catatan manual dapat menjadi tidak akurat jika tidak dicatat dengan benar oleh petugas yang bertanggung jawab. Oleh karena itu, perusahaan harus menetapkan sistem tata kelola yang baik untuk pencatatan produksi harian tanaman karet.

Android merupakan sistem operasi untuk *smartphone* yang menawarkan platform terbuka kepada pengembang untuk dapat membuat aplikasi mereka sendiri. Perkembangan *mobile phone* khususnya *smartphone* di Indonesia sangat pesat. Dari total populasi penduduk Indonesia yang mencapai 272,1 juta jiwa terdapat 345,3 juta pengguna *mobile phone* yang berarti banyak dari penduduk yang memiliki lebih dari satu perangkat *mobile phone*. Jumlah ini sudah bertambah 27 juta pengguna dari tahun 2020 di Indonesia. Kepemilikan perangkat *smartphone* dari penduduk Indonesia adalah yang terbesar diantara yang lain yaitu sebanyak 94%. Jumlah ini lebih banyak dari perangkat lainnya seperti laptop/komputer sebanyak 66% dan tablet 23%. Selain itu jika dicermati lebih dalam pada sistem operasi di *smartphone* maka yang paling sering digunakan di Indonesia adalah sistem operasi *android* yaitu sebanyak 93.2% dan untuk *iOS* dari *Apple* yang terdapat pada produk *iPhone* sebanyak 6.4% sedangkan untuk 0.2% lainnya merupakan sistem operasi lain [2].

Pemanfaatan *smartphone* untuk memonitor produksi adalah solusi efektif untuk mencatat hasil produksi. Ini memberikan banyak keuntungan, termasuk memungkinkan penyadap atau mandor produksi untuk memantau kondisi kebun karet dan mencatat data langsung ke dalam sistem terintegrasi. Metode ini juga mengurangi risiko kesalahan input data dan memungkinkan monitoring produksi secara real-time, sehingga memungkinkan pengambilan keputusan yang cepat dan akurat.

Pencatatan produksi harian dengan platform aplikasi *mobile* dapat meminimalisir terjadinya kehilangan atau kerusakan pada dokumen pencatatan produksi harian tanaman karet. Oleh karena itu, perlu dilakukan penelitian mengenai perancangan sistem monitoring dan pencatatan produksi harian tanaman karet berbasis *android*. Sistem monitoring ini diharapkan dapat menjadi solusi yang

efektif dan efisien pada permasalahan pencatatan secara manual pada kebun karet Unit Tebenan PT. Perkebunan Nusantara VII.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu bagaimana membangun sistem monitoring dan pencatatan produksi harian tanaman karet berbasis *android* untuk mandor Unit Tebenan PT. Perkebunan Nusantara dengan metode kanban dan menggunakan teknologi *framework flutter*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun sebuah sistem monitoring dan pencatatan produksi harian tanaman karet berbasis *android* untuk mandor Unit Tebenan PT. Perkebunan Nusantara.

1.4 Manfaat Penelitian

Manfaat Penelitian ini adalah sebagai berikut:

1. Tersedianya literatur mengenai perancangan sistem monitoring dan pencatatan produksi harian tanaman karet berbasis *android*.
2. Penelitian ini diharapkan memberikan manfaat dalam mempermudah mandor Unit Tebenan PT. Perkebunan Nusantara VII melakukan monitoring dan pencatatan produksi harian tanaman karet.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi yang dikembangkan khusus diterapkan untuk Unit Tebenan PT. Perkebunan Nusantara VII dan tidak diimplementasikan secara publik.
2. Pengembangan aplikasi hanya berfokus pada pengembangan Front-end pada aplikasi *android*.
3. Kebutuhan pengembangan aplikasi ditentukan oleh pihak PT. Perkebunan Nusantara VII.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada skripsi ini adalah pembagian menjadi 5 bab sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menguraikan secara umum mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada Bab ini membahas mengenai prinsip, dasar teori tanaman karet, PTPN 7, Monitoring hasil produksi, *Kanban Method*, *Android*, *Local Storage Android*, *API*, *Open weather API*, *Postman*, *GPS*, *JSON*, *Blackbox testing*, *CASE Tools (VS Code, Flutter, Git, Github, CRUD Booster, Trello)*, *state of the art*, dan penelitian terkait yang berfungsi sebagai sumber dalam memahami permasalahan mengenai pengembangan *Aplikasi Monitoring produksi harian mandor Unit Tebenan PT. Perkebunan Nusantara VII*.

BAB III : METODE PENELITIAN

Pada Bab ini membahas mengenai metode *kanban* yang terdiri dari *requirement gathering*, penentuan *user story*, *breakdown user story* menjadi *backlog (WOW)*, tahap *Work in Progress (WIP)*, *testing* dan evaluasi yang digunakan dalam pengembangan *Aplikasi Monitoring produksi harian mandor Unit Tebenan PT. Perkebunan Nusantara VII*.

BAB IV : HASIL DAN PEMBAHASAN

Pada Bab ini berisi tentang hasil serta pembahasan yang diperoleh dalam penelitian.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian dan saran-saran sebagai masukan untuk penelitian lanjutan di masa mendatang.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Tanaman Karet

Karet memiliki banyak kegunaan yang penting dalam kehidupan sehari-hari dan berbagai industri. Salah satu kegunaan karet yang paling terkenal adalah sebagai bahan dasar pembuatan ban kendaraan, baik roda mobil, sepeda motor, maupun pesawat terbang. Selain itu, karet juga digunakan dalam pembuatan produk karet lainnya, seperti karet gelang, alat bantu dengar, dan bantalan getaran. Di bidang industri, karet juga digunakan sebagai bahan isolasi suara dan getaran pada mesin dan alat-alat industri. Kegunaan karet yang beragam membuatnya menjadi komoditas yang penting dalam perekonomian global.

Karet (*Hevea brasiliensis* Muell. Arg) merupakan salah satu komoditas perkebunan yang penting baik untuk lingkup Indonesia maupun bagi internasional. Indonesia pernah menguasai produksi karet dunia dengan mengungguli produksi negara-negara lain. Tanaman karet merupakan salah satu komoditi perkebunan yang menduduki posisi cukup penting sebagai sumber devisa non migas bagi Indonesia. Luas areal karet Indonesia saat ini, 85% (2.8 juta ha) merupakan areal perkebunan karet rakyat yang memberikan kontribusi 81% terhadap produksi karet alam nasional. Secara keseluruhan, luas pertanaman karet di Indonesia dari tahun 1995 hingga tahun 2012 cenderung fluktuatif, pada tahun 2012 menunjukkan peningkatan yang cukup signifikan. Tahun 2012, Indonesia memiliki perkebunan karet seluas 3.506.201 ha dengan produksi mencapai 3.012.254 ton dan produktivitas 1.073 kg/ha. Pada tahun 2025, Indonesia menargetkan menjadi negara penghasil karet alam terbesar di dunia dengan produksi 3.8 - 4.0 juta ton per tahun [3].

2.2 PT. Perkebunan Nusantara VII

PT Perkebunan Nusantara VII adalah perusahaan perkebunan yang berbasis di Lampung, Indonesia. PTPN VII didirikan pada tahun 1996 sebagai hasil penggabungan dari enam perusahaan perkebunan di Jawa Tengah dan DIY. Saat ini, perusahaan ini memiliki luas lahan perkebunan seluas 45.482,61 hektar yang tersebar di beberapa wilayah di Lampung, seperti Kabupaten Lampung Selatan, Lampung Tengah, Lampung Timur, dan Way Kanan dan juga di wilayah lain yaitu sumatra selatan dan bengkulu. Jenis komoditas yang dihasilkan oleh perusahaan ini meliputi karet, kelapa sawit, teh, dan gula. Perusahaan ini memiliki visi untuk menjadi perusahaan perkebunan terkemuka di Indonesia yang berbasis pada prinsip-prinsip keberlanjutan [4].

2.3 Monitoring Hasil Produksi

Monitoring hasil produksi adalah proses rutin pengumpulan data dan pengukuran kemajuan atas hasil yang didapatkan. Memantau perubahan yang fokus pada proses dan keluaran. Monitoring menyediakan data dasar untuk menjawab permasalahan. Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses suatu objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis tindakan untuk mempertahankan manajemen yang sedang berjalan.

Secara umum monitoring bertujuan mendapatkan umpan balik bagi kebutuhan program proses pembelajaran yang sedang berjalan, dengan mengetahui kebutuhan ini pelaksanaan program akan segera mempersiapkan kebutuhan dalam pembelajaran tersebut. Kebutuhan bisa berupa biaya, waktu, personel, dan alat. Pelaksanaan program akan mengetahui berapa biaya yang dibutuhkan, berapa lama waktu yang tersedia untuk kegiatan tersebut. Dengan demikian akan diketahui pula

berapa jumlah tenaga yang dibutuhkan, serta alat apa yang harus disediakan untuk melaksanakan program tersebut [5].

2.4 *Kanban Method*

Kanban merupakan model pengembangan perangkat lunak yang berfokus untuk mengatur aliran kerja (*workflow*) dan informasi dalam pengembangan perangkat lunak. Kanban memiliki konsep utama yaitu *visualize workflow* atau memvisualisasikan alur kerja, *limit Work in Process (WIP)* atau membatasi proses dalam pekerjaan, dan *measure and improve flow* atau melakukan peningkatan aliran kerja.

Pemvisualisasian ini dilakukan menggunakan sebuah *kanban board* yang berisi proses-proses dalam pengembangan seperti perencanaan, perancangan, persetujuan, pengembangan, pengujian, pengintegrasian, dan penerapan. Membatasi pekerjaan atau *limit WIP* dilakukan dengan tujuan agar pekerjaan dilakukan lebih optimal. Pada dasarnya sebuah tim memiliki batasan maksimal pekerjaan yang dilakukan agar mencapai hasil yang optimal. Melakukan peningkatan aliran kerja dilakukan untuk mencapai estimasi waktu yang sesuai. Menemukan dan mengaplikasikan matriks pekerjaan yang sesuai merupakan langkah untuk melakukan pengestimasian pekerjaan dengan tepat [6].



Gambar 2.1. Workflow kanban

2.5 *Android*

Android adalah sistem operasi dan platform pemrograman yang dikembangkan oleh *Google* untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). *Android* bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda [7]. *Android* merupakan sistem open source sehingga para pengembang bebas melakukan modifikasi, pembaruan, dan pendistribusian.

Selain itu sifat open source ini membuat para pengembang aplikasi bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan *Google*. Selain itu sifat *open source* ini membuat para pengembang aplikasi bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan *Google*. Pengembangan aplikasi *android* menggunakan bahasa pemrograman *Java*, *Kotlin*, dan *Dart* (*Framework Flutter*).



Gambar 2.2. *Android Version*

2.5.1 *Local Storage Android*

Sistem file pada *android* serupa dengan sistem file pada platform berbasis disk lainnya. Sistem ini menyediakan beberapa opsi untuk menyimpan sebuah data pada aplikasi antara lain sebagai berikut [8] :

1. *App-specific storage* atau penyimpanan khusus aplikasi yang digunakan untuk menyimpan file yang hanya ditujukan untuk pengguna aplikasi.

2. Penyimpanan Bersama yaitu penyimpanan yang akan menyimpan file yang dapat dibagikan ke aplikasi lain.
3. *Preference* yaitu penyimpanan yang digunakan untuk menyimpan data-data primitif yang merupakan pasangan *key-value*.
4. *Database* yaitu basis data yang menyimpan data terstruktur. Library yang digunakan yaitu *Get Storage*.

Paket *get_storage* dalam *Flutter* adalah solusi penyimpanan ringan dan efisien yang memudahkan proses penyimpanan data *key-value* di penyimpanan lokal perangkat. Ini menawarkan *API* yang sederhana dan mudah digunakan untuk menyimpan dan mengambil data tanpa memerlukan pengaturan atau konfigurasi yang rumit. Dengan *get_storage*, pengembang dapat dengan cepat menyimpan berbagai jenis data seperti string, integer, boolean, atau bahkan daftar dan peta. Ini mendukung operasi sinkron dan asinkron, yang memungkinkan integrasi yang mulus dengan aplikasi *Flutter*. Baik itu menyimpan preferensi pengguna, data caching, atau mengelola status aplikasi, *get_storage* menyediakan solusi yang nyaman dan dapat diandalkan untuk kebutuhan penyimpanan lokal dalam proyek-proyek *Flutter* [9].

2.6 *Application Programming Interface (API)*

Application Programming Interface (API) adalah teknologi yang digunakan untuk melakukan pertukaran data dan informasi antara dua perangkat lunak atau lebih. Teknologi *API* ini digunakan secara luas karena kecepatan, kemudahan, dan portabilitas data yang dipertukarkan melalui *API* memungkinkan digunakan di teknologi modern manapun. *API* terbukti menjadi saluran data yang sempurna untuk pengembangan *Internet of Things (IoT)* dan *Mobile Apps* karena sifatnya yang ringan dan memiliki format pertukaran yang sederhana [10].

2.6.1 *Open Weather Map API*

Open Weather Map API adalah sebuah layanan *web* yang menyediakan informasi tentang kondisi cuaca, ramalan cuaca, dan data iklim untuk lokasi-lokasi di seluruh dunia. Layanan ini didirikan pada tahun 2012 dan memiliki data cuaca secara real-time, termasuk informasi tentang suhu, kelembapan, tekanan udara, arah dan kecepatan angin, curah hujan, dan indeks *UV*.

Layanan ini dapat diakses melalui situs webnya, serta melalui *API (Application Programming Interface)* yang tersedia untuk pengembang aplikasi dan situs web. Pengguna *API* dapat mengakses data cuaca yang terbaru dan juga ramalan cuaca untuk masa yang akan datang. Selain itu, *OpenWeatherMap* juga menyediakan data historis dan data iklim untuk area tertentu.

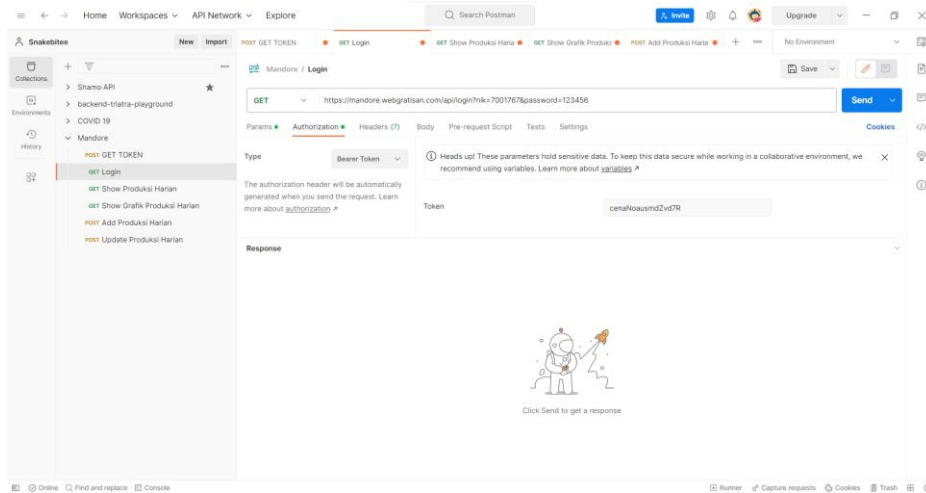
OpenWeatherMap sangat berguna bagi orang-orang yang memerlukan informasi tentang cuaca dan ramalan cuaca untuk keperluan bisnis, pariwisata, transportasi, olahraga, atau kepentingan pribadi. Data cuaca yang akurat dan terkini dari *OpenWeatherMap* dapat membantu pengguna dalam perencanaan kegiatan dan pengambilan keputusan [11].

2.6.2 *Postman API*

Postman API adalah sebuah platform pengembangan *API* yang digunakan untuk menguji, merancang, dan mendokumentasikan *API*. *Postman API* memungkinkan pengguna untuk membuat permintaan *API HTTP* dengan mudah dan mengatur responsnya dalam bentuk yang terstruktur. Selain itu, *Postman API* juga menyediakan fitur-fitur seperti pengujian otomatis, sinkronisasi kolaboratif, serta kemampuan untuk mengintegrasikan dengan alat pengembangan lainnya.

Pengguna *Postman API* dapat membuat permintaan *API* dengan mudah dan mengelola responsnya dalam format yang terstruktur. Selain itu, *Postman API* juga mendukung berbagai metode permintaan *HTTP*, seperti *GET*, *POST*, *PUT*, *DELETE*, dan lain-lain. Fitur-fitur lainnya yang disediakan oleh *Postman API* termasuk pengujian otomatis, penyimpanan lingkungan, serta integrasi dengan alat pengembangan lainnya seperti *Git*.

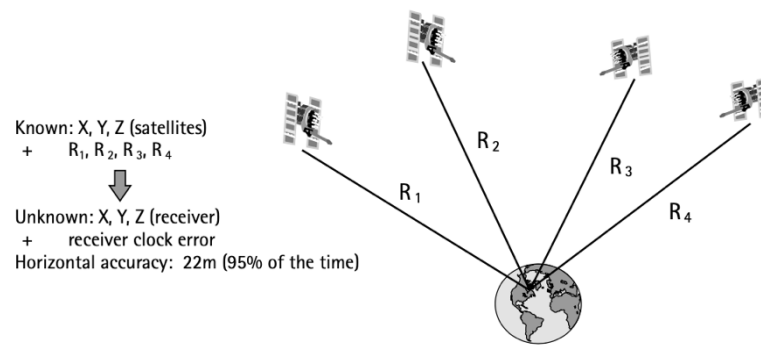
Postman *API* banyak digunakan oleh pengembang aplikasi dan situs web yang membutuhkan alat pengembangan yang mudah digunakan dan mendukung integrasi dengan banyak alat pengembangan lainnya. Platform ini juga tersedia dalam bentuk aplikasi *desktop* dan sebagai layanan berbasis web [12].



Gambar 2.3. *Workspace Postman*

2.7 *Global Positioning System (GPS)*

Global Positioning System merupakan sistem navigasi berbasis satelit yang pada mulanya dikembangkan oleh Departemen Pertahanan Amerika Serikat pada tahun 1970-an. *Global Positioning System* atau GPS menyediakan informasi penentuan lokasi dan waktu yang berkelanjutan dimanapun tanpa dipengaruhi oleh cuaca dan waktu. GPS menyediakan pemosisian dengan biaya efektif dengan akurasi yang tinggi. GPS telah diaplikasikan di berbagai bidang darat, laut dan udara [13]. Selain itu teknologi GPS ini tersedia pada *smartphone*. Hal ini memungkinkan untuk mengakses lokasi berdasarkan titik koordinat yang disesuaikan satelit menggunakan aplikasi yang ada pada *smartphone* tersebut.



Gambar 2.4. Cara kerja Global Positioning System (GPS)

Pada gambar 2.4. merupakan gambar yang menampilkan cara kerja GPS. Berdasarkan gambar tersebut maka untuk mengetahui posisi dari suatu *device* penerima maka dibutuhkan minimal empat buah satelit. Dari masing-masing satelit yang mengorbit di bumi kemudian mengirimkan data-datanya terutama data posisinya dari bumi ke penerima (*receiver*). Untuk mengetahui jarak satelit dari bumi maka digunakan perhitungan fisika yaitu lama transfer sinyal dari satelit ke *receiver* dikalikan dengan kecepatan transfer sinyal. Data posisi dan waktu pengiriman sinyal dari masing-masing satelit kemudian diterima dan digunakan untuk menentukan lokasi dari *receiver* atau penerima.

2.8 *Javascript Object Notation (JSON)*

Javascript Object Notation atau JSON merupakan sebuah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah dibuat (*generate*) dan diterjemahkan oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar *ECMA-262* Edisi ke-3 – Desember 1999.

JSON merupakan sebuah format penulisan yang tidak bergantung pada bahasa apapun. Hal ini karena JSON menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, *Java*, *JavaScript*, Perl, *Python*, dll. Karena sifat-sifatnya tersebut maka *JSON* menjadi format pertukaran data yang ideal [14].


```

1  {
2  "api_status": 1,
3  "api_message": "success",
4  "data": [
5    {
6      "id": 238,
7      "nik": "7001767",
8      "r_cl_kering": 12,
9      "r_lateks_kering": 18,
10     "r_hk": 7,
11     "tgl": "2022-09-01",
12     "cuaca": "Clouds",
13     "r_total_kering": 30,
14     "r_drc_lateks": 30.6,
15     "r_drc_cl": 28.6,
16     "luas_sadap": 50.4,
17     "r_kgHa": 0.5952380952380952,
18     "r_kgHk": 4.285714285714286
19   }
20 ]
21 }

```

Gambar 2.5. Contoh Format JSON

Pada gambar 2.3 menampilkan contoh format *Javascript Object Notation* atau *JSON* yang dapat dilihat bahwa format *JSON* di*API*t dengan kurung kurawal serta memiliki pasangan *key* dan *value* yang setiap nilainya baik *key* dan *value* tersebut di*API*t dengan tanda petik dua.

2.9 Blackbox testing

Pengujian *black-box* disebut juga dengan pengujian fungsional. Dalam pengujian fungsional seseorang tidak memiliki akses ke internal program dan hanya peduli dengan bagian yang diakses dari luar program. Seorang penguji menerapkan masukan ke program, mengamati hasil yang terlihat secara eksternal dari program, dan menentukan apakah hasil program adalah hasil yang diharapkan atau tidak diharapkan [15]. Beberapa keuntungan dan kerugian dari pengujian *black-box* adalah sebagai berikut [16]:

Keuntungan pengujian *black-box*:

1. Mensimulasikan penggunaan sistem yang sebenarnya.
2. Tidak butuh keahlian atau memahami struktur sistem.

Kekurangan pengujian *black-box*:

1. Berpotensi untuk melewatkan kesalahan *logic* pada sistem.
2. Berisiko terjadinya pengujian yang redundan atau berulang-ulang.

Pelaksanaan pengujian *black-box* dapat dilakukan dengan beberapa teknik, diantaranya adalah sebagai berikut:

a. Equivalence Class Partitioning Testing

Teknik desain uji yang membagi domain masukan dari sistem yang diuji menjadi beberapa kelas (atau kelompok) kasus uji yang memiliki efek serupa pada sistem. Teknik pengujian ini cocok digunakan saat sebuah domain masukan terlalu besar untuk semua elemennya untuk digunakan sebagai masukan uji. Sehingga domain masukan dapat dipartisi menjadi sejumlah subdomain yang terbatas untuk memilih masukan pengujian. Setiap subdomain dikenal sebagai kelas ekuivalen (EC), dan berfungsi sebagai sumber setidaknya satu input uji. Semua input di kelas yang sama memiliki efek yang sama pada sistem yang diuji.

Pedoman umum pembuatan partisi dari masukan data uji adalah sebagai berikut:

- Kondisi input menentukan arange $[a, b]$: Identifikasi satu EC untuk $a \leq X \leq b$ dan dua kelas lainnya untuk $X < a$ dan $X > b$ untuk menguji sistem dengan input yang tidak valid.
- Kondisi input menentukan sekumpulan nilai: Buat satu EC untuk setiap elemen dari set dan satu EC untuk anggota yang tidak valid. Misalnya, jika input dipilih dari sekumpulan N item, maka $N + 1$ EC akan dibuat: (i) satu EC untuk setiap elemen dari himpunan $\{M_1\}, \{M_2\}, \dots, \{M_N\}$ dan (ii) satu EC untuk elemen di luar himpunan $\{M_1, M_2, \dots, M_N\}$.

b. Boundary Value Analysis

BVA adalah teknik pengujian yang memilih elemen disekitar batas domain input sehingga data disekitar tepi atas dan bawah kelas baik didalam atau diluar domain input tercakup dalam kasus uji. Dalam praktiknya, *tester* dan *developer* cenderung mengabaikan kondisi batas. Akibatnya, *bug* atau cacat cenderung terkonsentrasi di dekat batas antara domain input. Oleh karena itu, data uji dipilih pada atau di dekat batas. Dalam hal ini, teknik *BVA* merupakan perluasan dan penyempurnaan dari teknik *Equivalence Partitioning Testing*. Panduan yang umum digunakan pada teknik *BVA* adalah sebagai berikut:

Jika domain input memiliki jangkauan yang spesifik maka susun skenario/kasus uji dengan mempertimbangkan titik batas domain input dan titik tepat di luar batas domain input. Sebagai contoh, jika domain input adalah di kisaran dari $-10.0 \leq X \leq 10.0$. Ini akan menghasilkan data uji yaitu $\{-9.9 - 10.0, -10.1\}$ dan $\{9.9, 10.0, 10.1\}$

c. Error Guesing

Teknik desain uji di mana pengalaman penguji digunakan untuk (i) menebak kemungkinan jenis dan lokasi kesalahan dalam sistem dan (ii) uji desain khusus untuk mengeksposnya. Mendesain kasus uji menggunakan teknik *error guesing* pada dasarnya didasarkan pada pengalaman penguji dengan kode yang mirip dengan penerapan yang sedang diuji.

d. All-pair / Pairwise Testing

Teknik desain uji dimana kasus-kasus pengujian merupakan semua kombinasi masukan sistem dengan n variabel masukan. Misalkan notasi $D(w)$ menunjukkan himpunan nilai untuk variabel sembarang w . Untuk tiga variabel yang diberikan X , Y , dan Z , kumpulan nilainya adalah sebagai berikut: $D(X) = \{\text{True}, \text{False}\}$, $D(Y) = \{0, 5\}$, dan $D(Z) = \{Q, R\}$. Jumlah total kasus uji semua kombinasi adalah $2 \times 2 \times 2 = 8$.

e. Decision Tables Testing

Decision Tables adalah sebuah teknik desain uji yang menguji kombinasi yang berbeda dari beberapa domain input yang setara. Teknik desain uji ini merupakan solusi dari beberapa teknik desain yang hanya terbatas pada satu domain input seperti *BVA* dan *Equivalence Partitioning*. Struktur umum dari table keputusan atau *decision table* adalah terdiri dari satu set kondisi atau penyebab yang dianggap sebagai masukan dan satu set efek atau hasil yang dianggap sebagai keluaran serta *rules* atau aturan yang menjadi nilai masukan. Nilai masukan yang mungkin adalah yes/true (Y/T), no/false (N/F), dan don't care (-).

2.10 *Computer Aided Software Engineering tools*

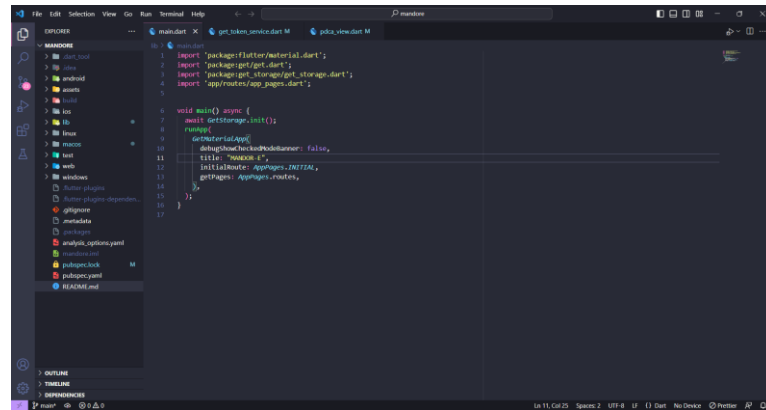
2.10.1 *Visual Studio Code*

Visual Studio Code (VS Code) adalah editor kode sumber yang dikembangkan oleh Microsoft dengan tujuan untuk memberikan pengalaman pengembangan yang efisien, kaya fitur, dan mudah digunakan. Editor ini terkenal karena kemampuannya untuk menyesuaikan antarmuka pengguna, memfasilitasi kolaborasi tim, dan menghadirkan banyak fitur terbaru untuk menulis dan mengedit kode. *VS Code* tidak hanya mendukung bahasa pemrograman populer seperti *JavaScript, Python, dan Java*, tetapi juga menawarkan dukungan untuk bahasa yang relatif kurang dikenal seperti *Go, Rust, dan R*.

VS Code menawarkan fitur *Intellisense* yang memudahkan pengembang dalam menulis kode sumber dengan cepat dan mudah. Fitur ini memungkinkan editor untuk memprediksi apa yang akan ditulis pengguna dan memberikan saran atau petunjuk yang sesuai secara otomatis. Selain itu, *VS Code* juga menyertakan debugger yang memungkinkan pengembang untuk melakukan debug pada kode sumber dan menemukan kesalahan atau bug yang terjadi.

VS Code juga menyediakan fitur seperti *Extension Marketplace* yang memungkinkan pengembang untuk mencari dan menginstal ekstensi atau plugin yang dapat meningkatkan kemampuan editor. Ekstensi ini memungkinkan pengembang untuk menyesuaikan dan memperluas kemampuan editor, mempercepat proses pengembangan, dan membuat pengembangan menjadi lebih efisien. *VS Code* juga dilengkapi dengan *Integrated Terminal* yang memungkinkan pengembang untuk menjalankan perintah *shell* dan memanggil aplikasi lain langsung dari dalam *VS Code*.

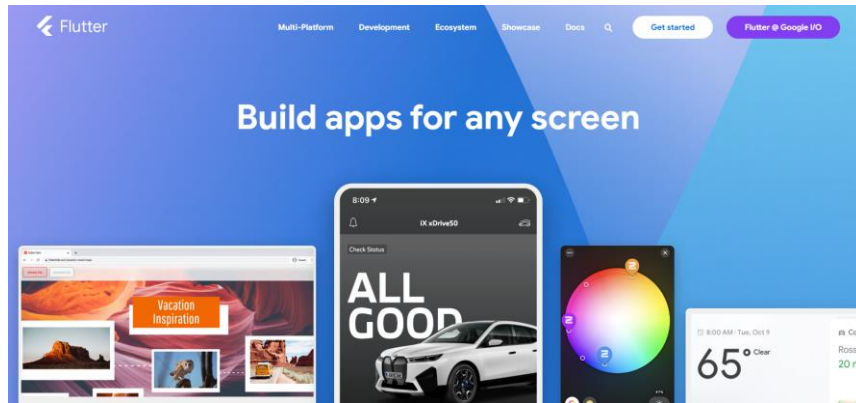
Secara keseluruhan, *VS Code* adalah editor kode sumber yang serbaguna, efisien, dan mudah digunakan. Pengembang perangkat lunak di seluruh dunia memuji editor ini karena dukungan bahasa yang luas, kemampuan *Intellisense* yang efektif, debugger yang andal, dan kemampuan untuk menyesuaikan editor sesuai kebutuhan. *VS Code* sangat berguna bagi pengembang yang ingin meningkatkan produktivitas mereka dan mempercepat proses pengembangan[17].

Gambar 2.6. *Workspace VS Code*

2.10.2 *Flutter*

Flutter adalah sebuah SDK atau framework open source yang dikembangkan oleh *Google* untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi *Android* dan *iOS*. *Flutter* menggunakan bahasa pemrograman *Dart* dalam pengkodean. Perbedaan framework *Flutter* dengan yang lainnya yaitu dalam build aplikasi, pada framework ini semua kodenya di compile dalam kode native-nya (*Android* NDK, LLVM, AOT compiled) tanpa ada interpreter pada prosesnya sehingga proses compile-nya menjadi lebih cepat [18].

Flutter adalah sebuah SDK atau framework *opensource* yang dikembangkan oleh *Google* untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi *Android* dan *iOS*. *Flutter* menggunakan bahasa pemrograman *Dart* dalam pengkodean. Perbedaan framework *Flutter* dengan yang lainnya yaitu dalam build aplikasi, pada framework ini semua kodenya di compile dalam kode native-nya (*Android* NDK, LLVM, AOT compiled) tanpa ada interpreter pada prosesnya sehingga proses compile-nya menjadi lebih cepat [19].



Gambar 2.7. Flutter

2.10.3 *Git*

Git adalah sistem manajemen versi yang digunakan oleh pengembang perangkat lunak untuk melacak perubahan pada kode sumber proyek. *Git* memungkinkan pengembang untuk bekerja secara terdistribusi, dengan setiap anggota tim dapat memiliki salinan kode sumber proyek yang sama.

Dalam *Git*, setiap perubahan pada kode sumber disimpan dalam sebuah commit, yang mencakup perubahan tertentu beserta catatan atau pesan yang menjelaskan perubahan tersebut. *Git* memungkinkan pengembang untuk memeriksa riwayat perubahan pada kode sumber, membandingkan versi kode sumber yang berbeda, dan kembali ke versi kode sumber tertentu jika diperlukan.

Git sangat populer dan banyak digunakan di seluruh dunia, terutama dalam proyek open-source. *Git* menyediakan banyak fitur yang mendukung proses pengembangan perangkat lunak, seperti sistem branch untuk bekerja pada fitur atau perbaikan tertentu, dan sistem merge untuk menggabungkan perubahan pada kode sumber [20].



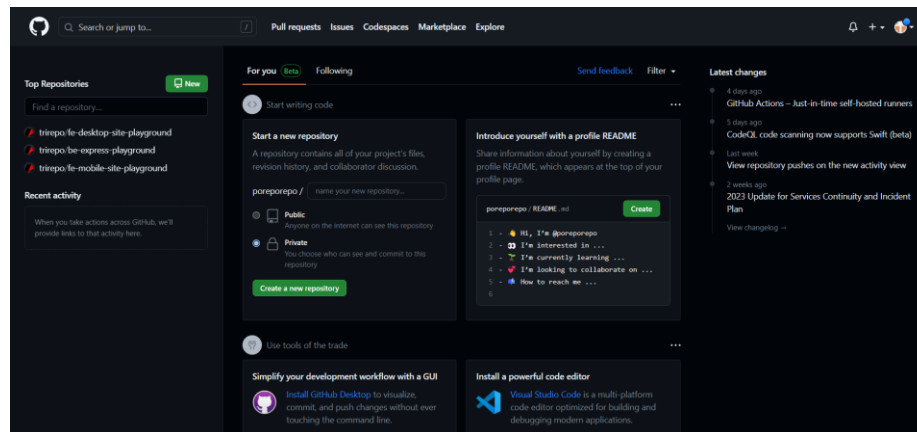
Gambar 2.8. Git

2.10.4 Github

GitHub adalah platform yang digunakan oleh pengembang perangkat lunak untuk berkolaborasi dalam proyek open-source dan menampung kode sumber dari proyek tersebut. *GitHub* memudahkan pengembang untuk menyimpan, membagikan, dan mengelola kode sumber secara terpusat sehingga memudahkan kolaborasi antara pengembang.

GitHub menyediakan banyak fitur yang mendukung proses kolaborasi, antara lain sistem manajemen versi, wiki, dan issue tracker. Dengan sistem manajemen versi, pengembang dapat dengan mudah melacak perubahan yang terjadi pada kode sumber proyek dan memperbaiki kesalahan yang terjadi. Wiki digunakan untuk membuat dokumentasi proyek dan memberikan informasi yang diperlukan bagi pengguna proyek. Issue tracker digunakan untuk melacak masalah yang terjadi pada proyek dan memungkinkan pengembang untuk berkolaborasi dalam menyelesaikan masalah tersebut.

GitHub sangat populer dan banyak digunakan di seluruh dunia, terutama dalam proyek open-source. *GitHub* memiliki banyak fitur yang mendukung kolaborasi dan memudahkan pengembang dalam menyimpan, membagikan, dan mengelola kode sumber proyek [21].

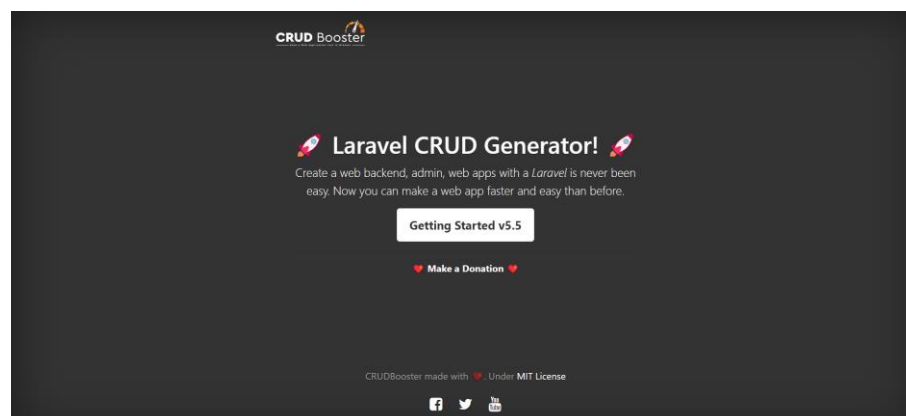


Gambar 2.9. GitHub

2.10.5 *CRUD Booster*

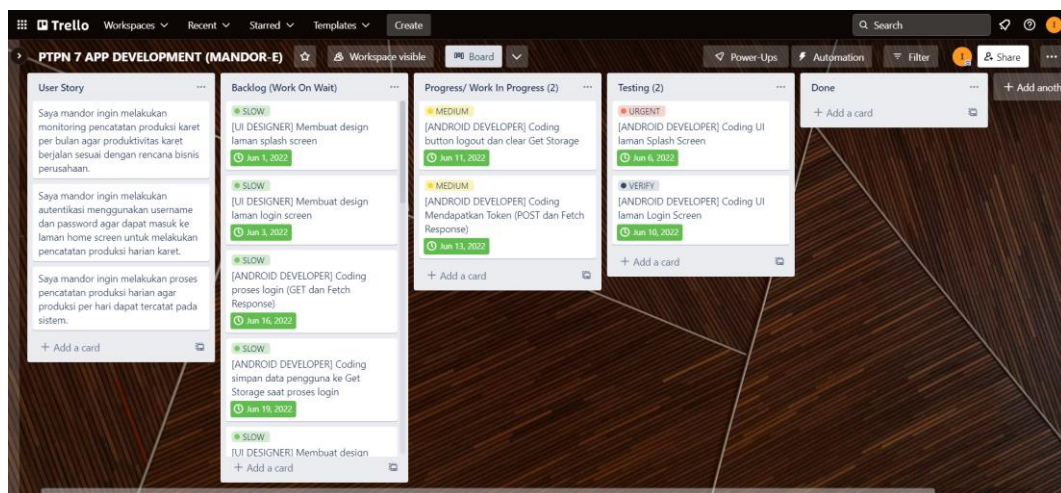
CRUD Booster adalah sebuah platform yang membantu pengembang perangkat lunak dalam membangun aplikasi berbasis web dengan cepat dan mudah. *CRUD* merupakan singkatan dari Create, Read, Update, dan Delete, yang merupakan operasi dasar dalam pengembangan aplikasi berbasis database. Dalam hal ini, *CRUD Booster* membantu pengembang dalam mempercepat proses pengembangan dengan menyediakan fitur-fitur dasar untuk operasi *CRUD* tersebut.

CRUD Booster menyediakan berbagai fitur, seperti generator kode otomatis, antarmuka pengguna yang responsif, manajemen data dan pengguna, serta integrasi dengan berbagai database dan bahasa pemrograman. Platform ini dirancang untuk memberikan solusi yang lebih mudah dan cepat bagi pengembang perangkat lunak dalam mengembangkan aplikasi web yang memiliki operasi *CRUD* [22].

Gambar 2.10. *CRUD Booster*

2.10.6 Trello

Trello sebuah alat yang digunakan sebagai papan proyek daring. *Trello* menyediakan cara mudah, gratis, fleksibel, dan visual untuk mengelola proyek dan mengatur apa pun didalamnya [23]. Manfaat dari penggunaan *trello* adalah dapat digunakan sebagai aplikasi kolaborasi yang memungkinkan semua anggota proyek tersebut bisa tahu apa yang sedang dikerjakan, siapa yang mengerjakannya, kapan pekerjaan tersebut direncanakan selesai, dan sejauh mana orang tersebut mengerjakannya.



Gambar 2.11. Kanban menggunakan *Trello*

2.11 Penelitian Terkait

Terdapat beberapa penelitian terkait yang dijadikan sebagai perbandingan serta rujukan mengenai metode serta hasil yang dicapai pada penelitian ini. Berikut merupakan ulasan dari beberapa penelitian terkait:

2.11.1 *Development and Application of Condition Monitoring System for Plant Production*

2.11.1.1 Deskripsi Penelitian

Penelitian yang telah dilakukan oleh SHENG Zhongqi, LIU Zhipeng, WANG Jianyong, LU Ying dengan judul “*Development and Application of*

Condition Monitoring System for Plant Production” membahas mengenai pengembangan dan aplikasi sistem pemantauan kondisi untuk produksi tanaman. Sistem pemantauan kondisi tersebut merupakan alat yang digunakan untuk memantau dan menganalisis kondisi lingkungan di sekitar tanaman, seperti suhu, kelembaban, pH tanah, dan konsentrasi nutrisi, yang berdampak pada pertumbuhan dan hasil panen tanaman. Sistem ini dirancang untuk memberikan informasi real-time dan akurat tentang kondisi lingkungan, sehingga dapat membantu petani untuk mengambil keputusan yang tepat dalam mengelola produksi tanaman.

Jurnal ini mungkin dapat bermanfaat bagi para petani dan peneliti yang tertarik dalam pengembangan sistem pemantauan kondisi untuk produksi tanaman. Dengan menggunakan sistem pemantauan kondisi yang akurat dan terintegrasi, petani dapat mengoptimalkan produksi tanaman dan meningkatkan hasil panen mereka [24].

2.11.1.2 Keterkaitan Penelitian

Penelitian yang dilakukan oleh SHENG Zhongqi, LIU Zhipeng, WANG Jianyong, LU Ying dengan judul “*Development and Application of Condition Monitoring System for Plant Production*” masih dapat ditingkatkan khususnya dibagian fitur akumulasi hasil produksi tanaman dan integrasi dengan *platform mobile*. Pada penelitian tersebut hasil penelitian masih menggunakan alat sebagai media monitoring produksi tanaman namun hasil penelitian belum melakukan integrasi dengan *platform* yang dapat mempermudah dalam melakukan monitoring produksi harian.

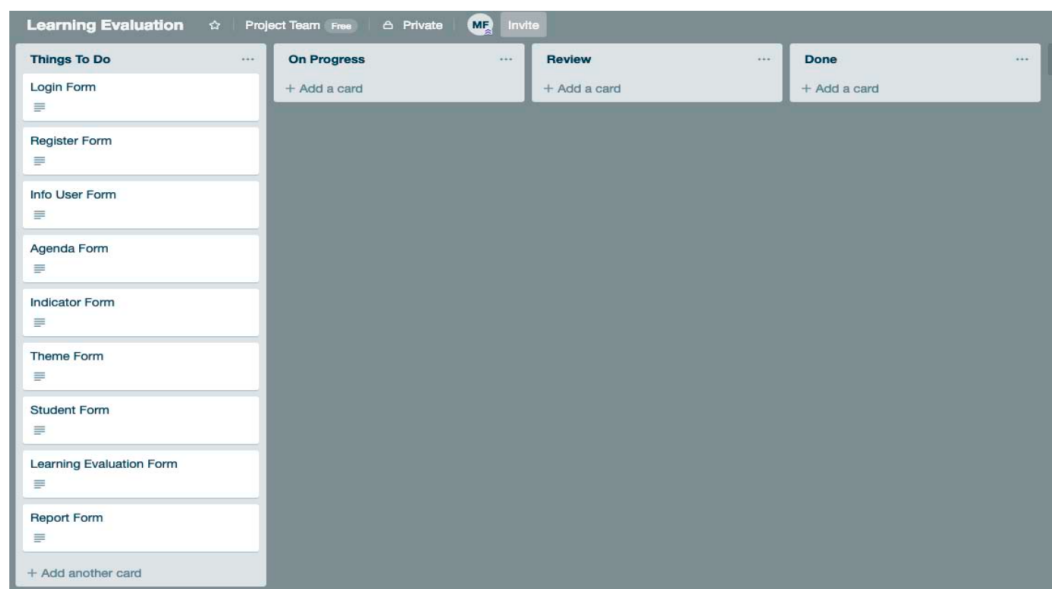
Hal inilah yang kemudian dikembangkan pada penelitian ini yaitu bahwa sistem monitoring tanaman dapat lebih lengkap dengan mengintegrasikan hasil penelitian yang dibuat dengan menggunakan *platform mobile*, sehingga dapat membantu meningkatkan efisiensi dan produktivitas tanaman.

2.11.2 Design of the Information System for Kindergarten Learning Evaluation used Kanban Methodology

2.11.2.1 Deskripsi Penelitian

Penelitian yang berjudul *Design of the Information System for Kindergarten Learning Evaluation used Kanban Methodology* yang dilakukan oleh M. Fitriawati dan R. H. Lestari dari Program Studi Sistem Informasi Universitas Komputer Indonesia pada tahun 2019 bertujuan untuk menghasilkan desain sistem informasi Pendidikan Anak Usia Dini (PAUD) yang membantu tenaga pendidik yaitu guru dalam mengevaluasi pembelajaran disetiap semester. Pengembangan yang dilakukan menggunakan metode Kanban yang bertujuan untuk membantu sinkronisasi tingkat produksi dengan menyesuaikan permintaan pelanggan yaitu guru.

Penelitian ini menjelaskan proses dasar pengerjaan atau pengembangan sistem menggunakan metode kanban.



Gambar 2.12. Kanban board yang digunakan dalam penelitian [25]

Metode kanban dijelaskan mulai dari *kanban board* yang dapat dilihat pada gambar 2.12 *Kanban board* yang digunakan dalam penelitian terkait ini terdiri dari empat kolom antara lain adalah sebagai berikut:

1. *Things to do* yaitu kolom yang berisi daftar dari *backlog* atau tugas yang harus dikerjakan.
2. *On Progress* yaitu kolom yang berisi tugas-tugas yang sedang dikembangkan.
3. *Review* yaitu kolom yang berisi daftar tugas yang berhasil dikembangkan dan dibutuhkan pengecekan kembali apakah telah sesuai dengan permintaan *customer* atau tidak.
4. *Done* yaitu kolom yang berisi tugas yang telah selesai.

Selain penjelasan *kanban board* dijelaskan juga proses visualisasi dari pengerjaan menggunakan *kanban board* tersebut.

2.11.2.2 Keterkaitan Penelitian

Penelitian yang dilakukan oleh M. Fitriawati dan R. H. Lestari menjelaskan mengenai model pengembangan *kanban* yang dilakukan. Penjelasan yang dilakukan pada penelitian ini cukup baik mulai dari *kanban board* hingga *visualisasi* pengerjaannya. Namun ternyata papan proyek yang digunakan kurang menggambarkan *kanban board* itu sendiri. Hal ini dikarenakan tidak terdapatnya batas *Work in Progress (WIP)* yang digunakan. Papan proyek yang tidak memiliki batas *WIP* cenderung seperti papan proyek *Scrum*. Selain itu penjelasan mengenai pembatasan *WIP* juga tidak disematkan dalam penjelasan penelitian tersebut.

Hal inilah yang kemudian dijelaskan dalam penelitian yang dilakukan yaitu mengenai *Rancang Bangun Aplikasi Monitoring Produksi Harian Mandor Unit Tebenan PT. Perkebunan Nusantara VII Berbasis Android*. Pada penelitian yang dilakukan digunakan metode pengembangan *kanban* dan dijelaskan mengenai *kanban board* yang digunakan dan pembatasan *Work in Progress (WIP)*.

2.11.3 Implementasi Metodologi Kanban Dalam Pembuatan Aplikasi E-Commerce Pertanian Dengan Pendekatan Zachman Framework

2.11.3.1 Deskripsi Penelitian

Penelitian yang dilakukan oleh Dicky Dewantoro, Condro Kartiko, Fauzan Romadlon pada Institut Teknologi Telkom Purwokerto berhasil mengembangkan aplikasi E-Commerce pertanian menggunakan pendekatan *zachman framework*. Fitur utama dari aplikasi ini adalah media data transaksi, disini pengguna dapat melakukan transaksi dengan lebih mudah oleh aplikasi e-tani pada gawai. Metode pengembangan yang digunakan adalah metode *kanban*.

| BACKLOG TASK & ID | STORY POINT | ORIGINAL ESTIMATE | D | D | D | D | D | STATUS |
|---------------------------------|-------------|-------------------|---|---|---|---|---|--------|
| | | | A | A | A | A | A | |
| | | | Y | Y | Y | Y | Y | |
| | | | 1 | 2 | 3 | 4 | 5 | |
| Minggu 1 | | | | | | | | |
| INITIAL PROJECT | 2 | 3 | 1 | 2 | 0 | 0 | 0 | DONE |
| TITLE LIBRARY | 2 | 4 | 1 | 3 | 2 | 0 | 1 | DONE |
| CONFIGURATI ON SERVICE | 4 | 2 | 2 | 1 | 4 | 0 | 0 | DONE |
| Minggu 2 | | | | | | | | |
| CREATE MODEL FOR REDUX | 4 | 2 | 1 | 3 | 3 | 1 | 0 | DONE |
| CREATE BUTTON LOGIN | 4 | 4 | 1 | 1 | 3 | 0 | 1 | DONE |
| CREATE BUTTON UPDATE | 4 | 4 | 2 | 1 | 2 | 0 | 1 | DONE |
| CREATE BUTTON JUAL | 4 | 3 | 1 | 1 | 1 | 0 | 1 | DONE |
| Minggu 3 | | | | | | | | |
| CREATE BUTTON PROFILE | 4 | 3 | 1 | 2 | 3 | 3 | 3 | DONE |
| CREATE BUTTON BERANDA BERAS | 4 | 3 | 3 | 1 | 3 | 2 | 1 | DONE |
| CREATE BUTTON KERANJANG | 4 | 3 | 2 | 3 | 2 | 1 | 2 | DONE |
| CREATE BUTTON TAMPILAN KATEGORI | 4 | 4 | 1 | 1 | 2 | 3 | 2 | DONE |

Gambar 2.13. *Kanban board* yang digunakan pada penelitian [26]

Penelitian terkait ini menggunakan *kanban board* seperti gambar 2.13 Informasi yang terdapat pada *kanban board* tersebut mencakup daftar *backlog*, *story point*, *original estimate*, hari, dan *status*. Tingkat kemajuan pada penelitian ini dilaporkan disetiap minggu mulai dari minggu pertama hingga minggu keenam.

Pengujian yang digunakan adalah pengujian *blackbox* dengan teknik *decision table testing*. Pengujian dilakukan pada tiga laman yaitu laman *login*, laman profil, dan laman formulir jual produk.

2.11.3.2 Keterkaitan Penelitian

Pada penelitian terkait ini yang telah dilakukan oleh Dicky Dewantoro, Condro Kartiko, Fauzan Romadlon pada Institut Teknologi Telkom Purwokerto terdapat beberapa hal yang masih dapat diperbaiki terutama pada bagian metode pengembangan dan pengujian. Metode pengembangan kanban khususnya pada *kanban board* yang digunakan terlihat tidak sesuai dengan *kanban board* pada umumnya yang terdiri dari kolom *Work on Wait (WOW)*, *Work in Progress (WIP)*, *Done*, sehingga pembaca akan merasa bingung dan memiliki banyak pertanyaan terkait *kanban board* yang digunakan. Selain itu tidak adanya batas atau limit *WIP* yang diterapkan dan penggunaan *story point* yang menjadi acuan dalam pengembangan. Hal ini tentu membuat pembaca bingung dan mengira bahwa ini semakin mendekati metode *scrum*.

Proses pengujian aplikasi menggunakan metode *blackbox* dan teknik *decision table testing* dilakukan untuk menguji kombinasi seluruh input yang tersedia pada laman. Hal ini tidaklah salah namun nyatanya dapat dilakukan penyederhanaan skenario sehingga proses pengujian lebih efektif.

Penelitian yang dilakukan mengenai *Rancang Bangun Aplikasi Monitoring Produksi Harian Mandor Unit Tebanan PT. Perkebunan Nusantara VII* meningkatkan hal-hal yang perlu ditingkatkan pada penelitian terkait ini.

2.12 State of The Art

State of the art adalah hasil analisis dari penelitian yang pernah dilakukan sebelumnya yang memiliki konsep sejalan serta berkaitan dengan sistem yang dikembangkan dalam penelitian ini. Dari hasil analisis tersebut maka dapat diketahui sejauh mana penelitian yang pernah dilakukan sebelumnya. Hal ini

kemudian dijadikan referensi pada penelitian ini serta digunakan untuk menghindari kesalahan yang pernah dilakukan pada penelitian sebelumnya. Penelitian ini mengambil enam referensi penelitian terdahulu yang telah dijelaskan pada poin *Penelitian Terkait*. Penelitian ini menggunakan beberapa referensi yang pernah dilakukan terutama dalam hal metode pencatatan produksi tanaman berbasis *smartphone android* dengan *workflow* atau alur kerja yang berbeda serta penambahan beberapa fitur yang berbeda dari penelitian sebelumnya.

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian dilakukan pada:

1. Waktu penelitian : April sampai dengan September 2022
2. Tempat penelitian : PT. Perkebunan Nusantara VII

3.2 Jadwal Penelitian

Jadwal pada penelitian ini adalah sebagai berikut:

Tabel 3.1. Jadwal Penelitian

| No | Aktivitas | Apr 2022 | May 2022 | Jun 2022 | Jul 2022 | Agu 2022 | Sep 2022 |
|----|-----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1 | Penentuan Kebutuhan | | | | | | |
| 2 | Penentuan <i>User story</i> | | | | | | |
| 3 | Perancangan | | | | | | |
| 4 | <i>Work On Wait (WOW)</i> | | | | | | |
| 5 | <i>Work in Progress (WIP)</i> | | | | | | |
| 6 | Pengujian | | | | | | |
| 7 | <i>Done</i> | | | | | | |
| 8 | Pelaporan | | | | | | |

3.3 Alat Dalam Penelitian

3.3.1 Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3.2. Alat yang digunakan dalam penelitian

| No | Nama Alat | Spesifikasi | Deskripsi |
|----|---------------------------|--|---|
| 1 | Laptop | <i>Intel Core i5, RAM 16GB, dengan sistem operasi windows 11</i> | Perangkat keras yang digunakan sebagai compiler dalam pemrograman |
| 2 | <i>Smartphone Android</i> | <i>Android 6 QKQ1 dan 11</i> | Perangkat keras sebagai tempat peng- <i>installan</i> aplikasi |
| 3 | <i>Visual Studio Code</i> | Versi 1.74.3 | <i>Integrated Development Environment (IDE)</i> untuk melakukan pengembangan aplikasi <i>smartphone</i> berbasis <i>android</i> |
| 4 | <i>Flutter</i> | Versi 3.0.1 | <i>Framework</i> untuk membuat aplikasi multiplatform |
| 5 | <i>Open Weather Map</i> | <i>Maps for Android</i> | <i>Maps</i> sebagai perantara untuk mengambil titik lokasi dan cuaca di <i>flutter</i> |
| 6 | <i>Get Storage</i> | Versi 2.0.3 | <i>Database</i> tempat penyimpanan data pada <i>android</i> . |
| 7 | <i>Figma</i> | Versi 28.4.12.3 | Perangkat lunak untuk melakukan perancangan antarmuka |
| 8 | <i>CRUD Booster</i> | Versi 5.5 | Perangkat lunak untuk menggenerasi <i>API</i> dan mengelola admin |

Tabel 3.2. Alat yang digunakan dalam penelitian (lanjutan)

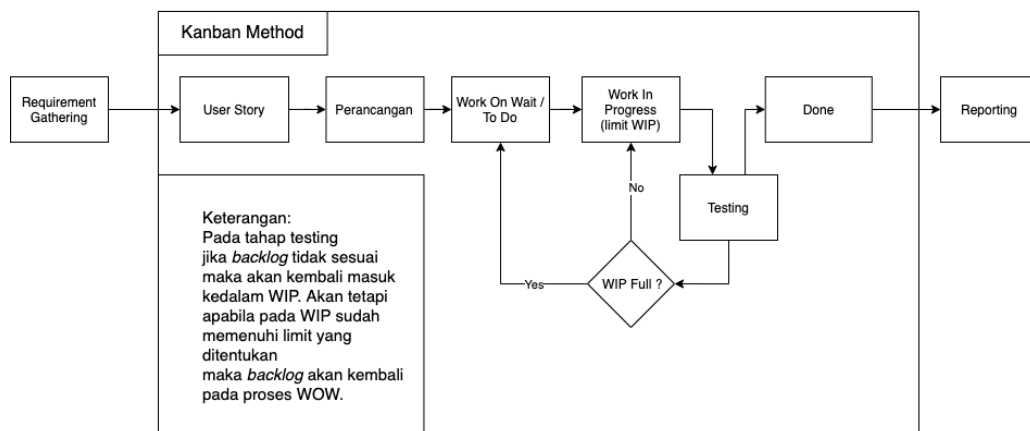
| No | Nama Alat | Spesifikasi | Deskripsi |
|----|------------------------|-----------------------------|--|
| 9 | Niaga Hoster | Website Hosting | Tools untuk melakukan hosting API |
| 10 | Trello | Online Trello | Tools yang digunakan sebagai <i>kanban board</i> untuk memvisualisasi alur kerja |
| 11 | Lingkungan Operasional | Android OS version 6 dan 11 | OS Version yang digunakan untuk implementasi aplikasi |

3.3.2 Bahan Penelitian

Bahan yang digunakan dalam penelitian ini adalah data produksi tanaman karet harian pada area tanaman karet Unit Tebenan PT. Perkebunan Nusantara VII.

3.4 Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini mengikuti model pengembangan perangkat lunak *Kanban*. Berikut merupakan gambar dari tahapan pengembangan menggunakan metode *Kanban*:

Gambar 3.1. Tahapan pada metode *Kanban*

Gambar 3.1. menggambarkan tahap-tahap dalam metode *Kanban*, dimulai dengan pengumpulan informasi sebelum pengembangan dimulai yang disebut *requirement gathering*. Kemudian dilanjutkan dengan tahap pengembangan sistem yang dimulai dengan menentukan *user story*, latar belakang masalah, dan motivasi dalam penelitian untuk menentukan fitur dan aktivitas pada aplikasi. Setelah menentukan *user story*, mereka akan dibagi menjadi beberapa *backlog* yang akan digunakan sebagai *task card* yang akan dikerjakan dan diletakkan pada *Work On Wait (WOW)*. Ketika *WOW* selesai, developer akan memulai kerja dengan menarik *task card* dari *WOW* ke *Work in Progress (WIP)*. Setelah selesai, *task card* akan dilakukan *testing* menggunakan metode *blackbox testing*. Jika berhasil, *task card* akan dipindahkan ke bagian *Done*, namun jika gagal, akan dipindahkan kembali ke *WIP*, asalkan jumlah task pada *WIP* belum mencapai batas maksimal yaitu 2 *task card*. Penjabaran dari masing-masing langkah dalam metode *Kanban* adalah sebagai berikut:

3.4.1 Target Pengguna

Tahap ini dilakukan pendefinisian mengenai pengguna yang menggunakan aplikasi serta tujuan pengembangan aplikasi. Studi kasus penelitian ini dilakukan di PT. Perkebunan Nusantara VII dengan *user* utama yang merupakan pengguna aplikasi yaitu mandor. Mandor merupakan seseorang yang melakukan kegiatan pencatatan produksi harian dari setiap penyadap karet pada area tertentu terutama pada unit yang memproduksi karet di PT. Perkebunan Nusantara VII.

3.4.2 Tahap Penentuan *User story*

Tahap ini dilakukan untuk menghasilkan sebuah daftar *user story* yang kemudian menjadi pedoman dalam penentuan *backlog* yang digunakan dalam pengembangan aplikasi. Hal ini dikarenakan *backlog* merupakan hasil turunan atau *breakdown* yang didapatkan dari *user story* itu sendiri. Format penggunaan *user story* yang digunakan dalam penelitian ini adalah sebagai berikut:

| |
|---|
| Saya [pengguna] ingin melakukan [kegiatan] agar [hasil yang diharapkan] |
|---|

Gambar 3.2. Format *user story* yang digunakan dalam penelitian

Gambar 3.2. merupakan format dari *user story* yang digunakan dalam penelitian. Pengguna yang diisikan merupakan target pengguna yang telah ditentukan dalam penelitian. Kegiatan adalah hal yang dilakukan oleh pengguna agar dapat mencapai hasil yang diharapkan melalui penggunaan aplikasi yang dikembangkan.

Tabel 3.3. Daftar *user story*

| No | User | User story |
|----|--------|---|
| 1. | Mandor | <ol style="list-style-type: none"> 1. Saya mandor ingin melakukan otentikasi menggunakan <i>username</i> dan <i>password</i> agar dapat masuk ke laman <i>home Screen</i> untuk melakukan pencatatan produksi harian karet. 2. Saya mandor ingin melakukan proses pencatatan produksi harian agar produksi per hari dapat tercatat pada sistem. 3. Saya mandor ingin melakukan monitoring pencatatan produksi karet per bulan agar produktivitas karet berjalan sesuai dengan rencana bisnis perusahaan. |

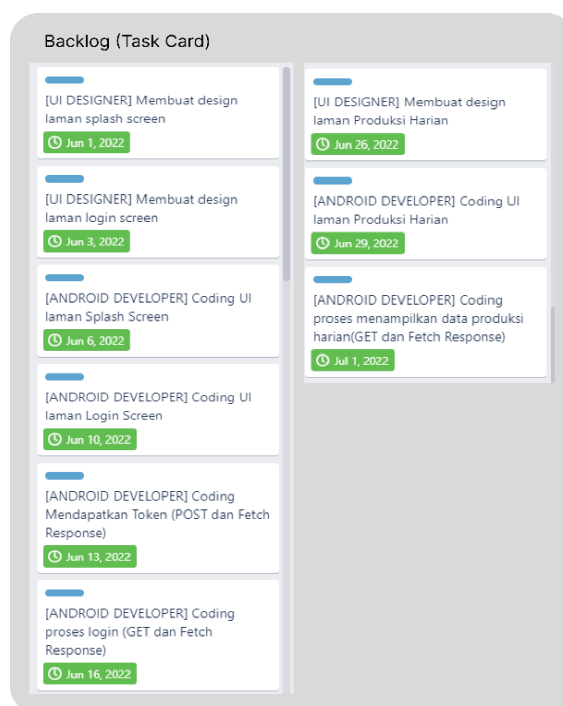
3.4.3 Tahap Perancangan

3.4.3.1 Perancangan Sistem

Tahap perancangan sistem dilakukan dengan melakukan *breakdown* dari *user story* yang telah ditentukan sebelumnya hingga menjadi beberapa *backlog*. Kemudian dari *backlog* tersebut dilakukan *breakdown* kembali hingga menjadi rincian-rincian. Rincian yang sama kemudian disatukan menjadi satu pekerjaan. Setelah persiapan *user story* selesai maka selanjutnya dilakukan proses perancangan sistem yang menentukan hal-hal sebagai berikut:

Penentuan *backlog*

Semua *user story* yang telah didefinisikan sebelumnya kemudian dilakukan *breakdown* yang kemudian menjadi sebuah *backlog*. *Backlog* yang didapat kemudian diletakkan sebagai *taskcard* pada *kanban board* menggunakan *tools trello*. *Backlog* atau *taskcard* inilah yang kemudian digunakan dalam proses pengembangan.

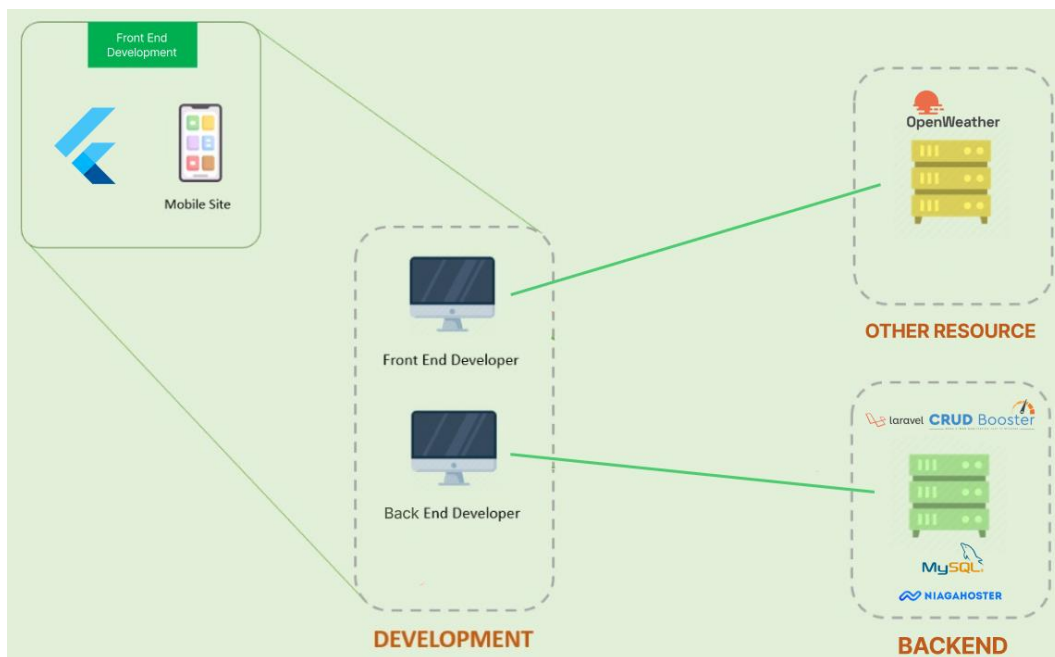


Gambar 3.3. Task card

Gambar 3.3. merupakan *breakdown* dari *user story* yang telah ditentukan sebelumnya hingga menjadi beberapa *backlog*. Kemudian dari *backlog* yang didapat diletakkan sebagai *taskcard* pada *kanban board* menggunakan *tools Trello*.

Penentuan *environment*

Struktur system project ini ditentukan berdasarkan framework ataupun aplikasi yang digunakan sesuai dengan kebutuhan pengembangan project aplikasi. Beberapa *framework* dipilih sebagai program untuk membuat aplikasi ini, adapula resource lain yang digunakan dalam pengembangan project. Berikut *environment* dalam mengembangkan project aplikasi ini.



Gambar 3.4. *Environment System*

Gambar 3.4. merupakan *environment* sistem untuk mengembangkan project aplikasi. Struktur tersebut sudah sesuai dibuat berdasarkan sistem yang dipilih untuk mengembangkan aplikasi ini.

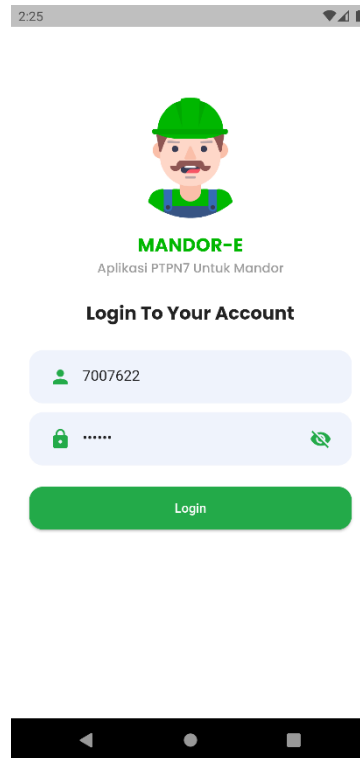
3.4.3.2 Perancangan Antarmuka

Perancangan antarmuka meliputi pembuatan antarmuka dari setiap laman yang ditampilkan pada aplikasi. Tahap perancangan dimulai ketika perancangan sistem telah selesai dilakukan. Hal ini dikarenakan untuk menghindari pembuatan kode yang berulang sehingga dalam perancangan sistem harus ditetapkan secara pasti tanpa ada lagi perubahan saat proses pengembangan dimulai. Perancangan antarmuka dilakukan menggunakan *software Figma*.



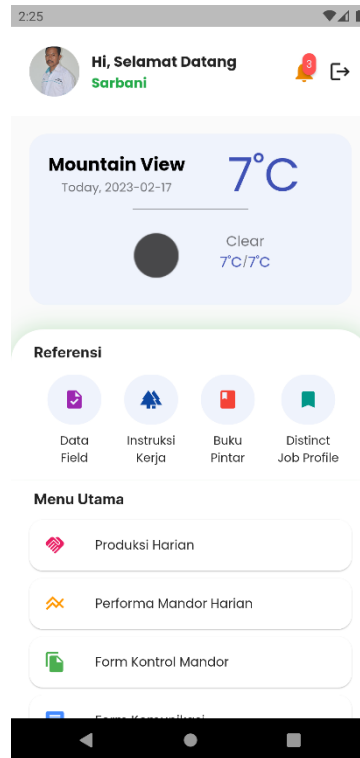
Gambar 3.5. *Splash Screen*

Gambar 3.5. merupakan desain antarmuka *splash Screen* yang dihasilkan dari *Figma*. Antarmuka yang dirancang ditampilkan dalam bentuk layar *smartphone* dengan orientasi *potrait*.



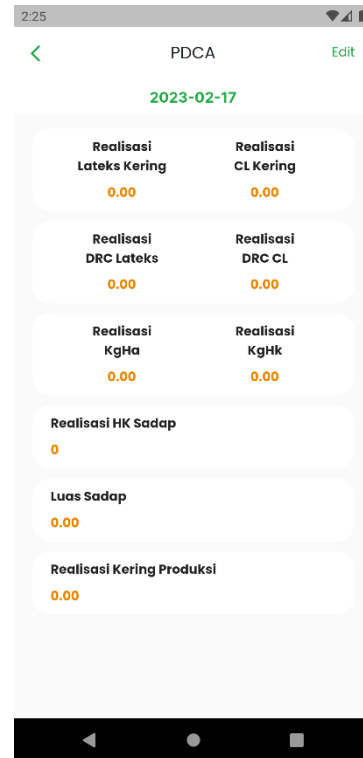
Gambar 3.6. *Login Screen*

Gambar 3.6. merupakan desain antarmuka *login Screen* yang dihasilkan dari *Figma*. Antarmuka yang dirancang menampilkan form yang berisi *username*, *password* serta tombol *login*.



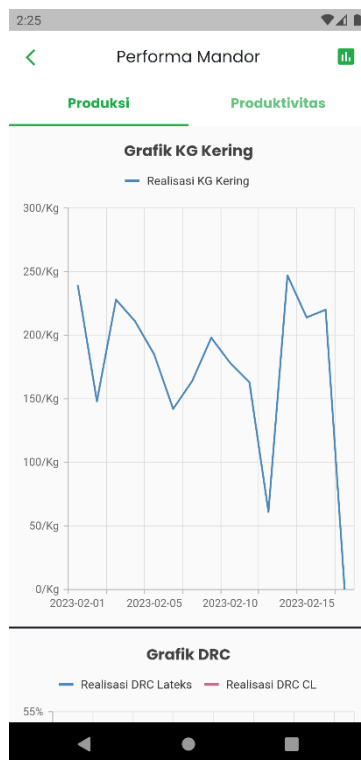
Gambar 3.7. *Home Screen*

Gambar 3.7. merupakan desain antarmuka *home Screen* yang dihasilkan dari *Figma*. Antarmuka yang dirancang menampilkan halaman *home* yang terdapat beberapa menu di dalamnya.



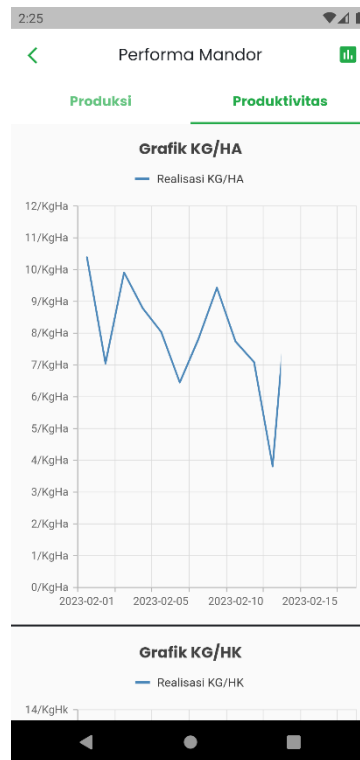
Gambar 3.8. Produksi Harian *Screen*

Gambar 3.8. merupakan desain antarmuka produksi harian *Screen* yang dihasilkan dari *Figma*. Antarmuka yang dirancang menampilkan jumlah produksi karet harian.



Gambar 3.9. Performa Mandor *Screen*

Gambar 3.9. merupakan desain antarmuka performa mandor *Screen* yang dihasilkan dari *Figma*. Antarmuka yang dirancang menampilkan grafik produksi per bulan.



Gambar 3.10. Performa Produktivitas Mandor *Screen*

Gambar 3.10. merupakan desain antarmuka produktivitas mandor *Screen* yang dihasilkan dari *Figma*. Antarmuka yang dirancang menampilkan grafik pada menu produktivitas per bulan.

3.4.4 Tahap Pengembangan Sistem

Proses ini dilakukan pengkodean dari sistem yang telah dirancang. Proses perancangan sistem yang telah dilakukan harus ditranslasikan kedalam kode. Hasil dari proses ini adalah suatu aplikasi yang siap untuk dilakukan pengujian atau *testing* menggunakan metode *Blackbox testing*. Sesuai dengan metode Kanban maka dalam proses pengembangan menggunakan beberapa prinsip diantaranya adalah sebagai berikut:

1. Memvisualisasi Alur Kerja
Proses visualisasi alur merupakan penggambaran alur pengembangan pada sebuah *kanban board*. Dalam penelitian ini sendiri *kanban board* digambarkan menggunakan *tools* bernama *Trello*.
2. Membatasi *Work in Progress* atau pekerjaan yang sedang berjalan
Dalam proses pengembangan aplikasi ini jumlah *Work in Progress* pada *kanban board* dibatasi sebanyak 2 jumlah *taskcard*.
3. Fokus pada *WIP*
Fokus pada *WIP* disini merupakan sebuah prinsip dari metode Kanban dimana tidak boleh terdapat suatu pekerjaan baru yang dilakukan sebelum pekerjaan yang lama terselesaikan.

3.4.5 Tahap Testing

Pengujian atau *testing* yang dilakukan menggunakan metode *Blackbox testing*. Pengujian dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi sehingga dapat dilakukan perbaikan. Pengujian dilakukan dengan *blackbox testing* atau pengujian secara langsung pada *smartphone* yang telah terinstal aplikasi. Teknik pengujian *blackbox* yang digunakan adalah menggunakan *decision table testing*. *Blackbox testing* atau pengujian secara langsung pada *smartphone* ini dilakukan dengan cara mengecek aplikasi terhadap *use case* yang telah dirancang. Hasil akhir dari pengujian adalah bahwa aplikasi telah bebas dari kesalahan atau *bug* sehingga diharapkan saat sampai ke tangan pengguna aplikasi sudah siap digunakan.

3.4.6 Tahap Analisis

Analisis dilakukan mengenai hasil dari pengujian mengenai pengembangan Aplikasi Monitoring Produksi Harian Mandor Unit Tebanan PT. Perkebunan

Nusantara VII Berbasis Android yang dilakukan setelah proses pengujian telah selesai dan data pengujian telah terkumpul.

3.4.7 Tahap Pelaporan

Tahap akhir dari penelitian ini adalah pelaporan hasil dan temuan penelitian mengenai *Aplikasi Monitoring Produksi Harian Mandor Unit Tebenan PT. Perkebunan Nusantara VII Berbasis Android*. Dari data yang dihasilkan dan telah dianalisis kemudian dilakukan pengambilan kesimpulan dan saran. Hasil temuan yang ada kemudian digunakan sebagai skripsi pada Universitas Lampung.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Kesimpulan yang diperoleh berdasarkan hasil dari penelitian adalah sebagai berikut:

1. Berhasil dikembangkannya sebuah aplikasi *android* yang menyediakan proses pencatatan dan monitoring produksi harian tanaman karet dan dilengkapi fitur otentikasi berbasis *android* untuk mandor Unit Tebenan PT. Perkebunan Nusantara.
2. Proses pengembangan aplikasi menggunakan 3 *user story* yang kemudian dilakukan *breakdown* hingga menghasilkan 21 *backlog*. *Backlog* ini kemudian disusun dalam *taskcard* serta diletakkan pada *kanban board* dan menjadi acuan dalam proses pengembangan hingga proses pengujian aplikasi.
3. Hasil pengujian menggunakan metode *blackbox testing* dengan 12 skenario didapatkan bahwa hasil pengujian yang dihasilkan telah sesuai dengan hasil yang diharapkan. Aplikasi berhasil berjalan pada *Android 6 Marshmallow* atau diatasnya sampai pada *android 11* dan telah sesuai dengan kebutuhan fungsional serta kebutuhan non-fungsional yang telah didefinisikan.

5.2 Saran

Saran yang dapat diberikan untuk melanjutkan penelitian ini berdasarkan hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Melakukan pengembangan aplikasi lebih lanjut dengan menambahkan fitur rekapitulasi produksi karet secara berkala seluruh mandor agar dapat mengetahui produksi karet secara keseluruhan.
2. Melakukan pengembangan aplikasi lebih lanjut dengan menambahkan fitur *tier rank* dan *poin* agar mandor mendapatkan apresiasi dari hasil pencatatan data hasil produksi karet pada aplikasi.
3. Melakukan *update* dari *dependency* atau *library* yang digunakan pada aplikasi sehingga sistem lebih *up to date*.
4. Melakukan pengembangan untuk versi *iOS* sehingga dapat menjangkau mandor karet Unit Tebenan PT. Perkebunan Nusantara VII yang menggunakan perangkat *iPhone* dan *iPad*.

DAFTAR PUSTAKA

- [1] PT. Perkebunan Nusantara VII, “Annual Report 2020”. Bandar Lampung, 2020.
- [2] We Are Social & Hootsuite, “Digital Data Indonesia 2020,”. Diakses pada: Jan 28, 2023. [Online]. Tersedia: <https://datareportal.com/reports/digital-2020-indonesia>.
- [3] R. N. Anwar and D. Suwanto, “Crop Management Rubber (*Hevea brasiliensis* Muell. Arg) in North Sumatra with Special Aspects of Nursery”. *Bul. Agrohorti* vol. 4, no. 1, pp. 94-103, 2016.
- [4] PT Perkebunan Nusantara VII, “Company Profile”. Bandar Lampung, 2020.
- [5] C. Pats Yahwe, L. Fid Aksara, “Rancang Bangun Prototype System Monitoring Kelembaban Tanah Melalui Sms Berdasarkan Hasil Penyiraman Tanaman ‘Studi Kasus Tanaman Cabai Dan Tomat’”. *J. Teknik Informatika, Universitas Halu Oleo*, vol. 2, no. 1, 2016.
- [6] P. Klipp, “Getting Started with Kanban”. California: CreateSpace Independent Publishing Platform, 2014.
- [7] G. Developer Training Team, “*Android Developer Fundamentals Course-Concept Reference*”. Diakses pada: Jan 28, 2023. [Online]. Tersedia: <http://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/RSADistributionTariffCode>

- [8] Developers *Android*, “Ringkasan penyimpanan data dan file”. 2020. Diakses pada: Jan 28, 2023. [Online]. Tersedia: <https://developer.android.com/training/data-storage>.
- [9] Flutter, “get_storage | *Flutter* Package”, 2023. Diakses pada: Mei 21, 2023. [online]. Tersedia: https://pub.dev/packages/get_storage.
- [10] D. Ashby and C. T. Jensen, John Wiley & Sons, “*APIs for Dummies*”. Hoboken, 2018.
- [11] WeatherMap Open “About company - OpenWeatherMap”. 2023, Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://openweather.co.uk/about>.
- [12] Postman, “Postman API App for *Desktop*” 2023, Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://www.postman.com/platform/api-platform>
- [13] A. El-Rabbany, “*Introduction to GPS: The Global Positioning System*”, London: Artech House, 2018.
- [14] D. Crockford, “Introducing JSON”, 1999. Diakses pada: Jun 24, 2020. [Online]. Tersedia: <https://www.json.org/json-en.html>.
- [15] D. DeVolder, S. Ghazanshahi, and J. Zadeh, “*Software testing and quality assurance*”, Western Illinois University, Macomb, 2008.
- [16] W. E. Perry, “*Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates*”. Amerika Serikat, 2007.
- [17] Microsoft. “*Visual Studio Code - Code Editing. Redefined*”, Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://code.visualstudio.com/>
- [18] F. Enggar Krisnada and R. Tanone, “Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis Mobile menggunakan *Flutter*”. *Jurnal T. Informatika dan Sistem Informasi*, Universitas Kristen Satya Wacana, vol. 5, no. 3, 2019.

- [19] M. H. Muladi, T. Listyorini, dan E. Supriyati, “Implementasi Augmented Reality Pada Pengenalan Tanaman Herbal Berbasis *Android*”. *J. Manajemen Informatika dan bisnis digital*, vol. 1, no. 2, 2022.
- [20] Linus Torvalds, “Git for *desktop*.”, 2023. Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://git-scm.com/>
- [21] GitHub, Inc, “GitHub”, 2023. Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://github.com/>
- [22] CRUDBooster Crocodic, “CRUDBooster - Laravel *CRUD* Generator”, 2023. Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://CRUDbooster.com/>
- [23] Atlassian, “About | What is Trello?”, 2023. Diakses pada: Feb 21, 2023. [Online]. Tersedia: <https://trello.com/>
- [24] Z. SHENG, Z. LIU, Jianyong, W. Ying LU, “Development and Application of Condition Monitoring System for Plant Production”. School of Mechanical Engineering & Automation, Northeastern University, *China*, 2012.
- [25] M. Fitriawati and R. H. Lestari, “Design of the Information System for Kindergarten Learning Evaluation used Kanban Methodology”, *D. Sistem Informasi*, vol. 662, 2019, doi:10.1088/1757-899X/662/2/022025
- [26] F. Dewantoro, D., Kartiko, C. and Romadhon, “Implementasi Metodologi Kanban Dalam Pembuatan Aplikasi E-Commerce Pertanian dengan Pendekatan Zachman Framework”. *F. Informatika*, vol. 5, no. 2, 2020.