**RANCANG BANGUN SISTEM *INVENTORY* BARANG PADA POLITEKNIK NEGERI INDRAMAYU BERBASIS *MOBILE* DENGAN MENGGUNAKAN METODE *AGILE***

**SKRIPSI**

Sebagai salah satu syarat untuk menyelesaikan Program Diploma IV Rekayasa Perangkat Lunak Politeknik Negeri Indramayu



**Oleh:**

**MAULANA HIDAYATULLAH**

**NIM 1905017**

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI INDRAMAYU**

**JANUARI 2023**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

|  |  |
| --- | --- |
| Nama | : Maulana Hidayatullah |
| NIM | : 1905017 |
| Program Studi | : Diploma IV Rekayasa Perangkat Lunak |
| Judul | : Rancang Bangun Sistem *Inventory* Barang Pada Politeknik Negeri Indramayu berbasis *Mobile* dengan menggunakan Metode *Agile* |
| Pembimbing | : Iryanto, S.Si., M.Si  NIP 199008012019031014 ........................... |

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal Agustus 2023 dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Terapan Program Studi Diploma IV Rekayasa Perangkat Lunak, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

|  |  |  |
| --- | --- | --- |
| Ketua Penguji | : A. Lubis Ghozali, S.Kom., M.Kom  NIP 198605102019031011 | ........................... |
| Anggota  Penguji I | : Riyan Farismana, S.Kom., M.Kom  NIP 198905112022031005 | ........................... |
| Anggota  Penguji II | : Iryanto, S.Si., M.Si  NIP 199008012019031014 | ........................... |

Indramayu, Agustus 2023

Ketua Jurusan Teknik Informatika

Eka Ismantohadi, S.Kom., M.Eng.

NIP 198107092021211005

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan dengan sesungguhnya bahwa **Skripsi** ini adalah asli hasil karya saya sendiri serta **Skripsi** ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar **Sarjana Terapan** di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam naskah ini dan disebutkan dalam daftar Pustaka

Indramayu, Agustus 2023

Yang menyatakan,

**Maulana Hidayatullah**

NIM. 1905017

ABSTRAK

Teknologi informasi telah berkembang pesat dalam hal implementasi, saat ini. Hal ini dapat diterapkan pada berbagai bidang seperti sistem manajemen barang. Saat ini, sistem pengelolaan barang atau *Inventory* di Politeknik Negeri Indramayu masih menggunakan sistem manajemen konvensional. Oleh karena itu, pengelolaan barang yang ada di kampus kurang optimal karena masih menggunakan proses manual. Oleh karena itu, penelitian ini mengusulkan Aplikasi *Inventory* Barang Pada Politeknik Negeri Indramayu berbasis *Mobile* telah memberikan kemudahan untuk pengguna dalam melakukan pengelolaan barang. Dengan adanya aplikasi ini, pengguna dapat mempermudah peminjaman, pemindahan, dan pengecekan barang. Implementasi metode *Agile* dapat menjadi pendekatan yang bermanfaat dalam perancangan Aplikasi *Inventory* Barang Pada Politeknik Negeri Indramayu berbasis *Mobile,* karena dapat menghasilkan aplikasi yang lebih sesuai dengan kebutuhan instansi Politeknik Negeri Indramayu, dan dapat meningkatkan pengelolaan barang pada Instansi Politeknik Negeri Indramayu serta lebih cepat dan lebih efisien bagi pengguna Aplikasi.

**Kata Kunci**: Politeknik Negeri Indramayu. Sistem *Inventory, Mobile.*

ABSTRACT

*Information technology has developed rapidly in terms of implementation, nowadays. This can be applied to various fields such as goods management systems. Currently, the goods or Inventory management system at Indramayu State Polytechnic still uses a conventional management system. Therefore, the management of goods on campus is less than optimal because it still uses a manual process. Therefore, this research proposes a Mobile-based Indramayu State Polytechnic Inventory Application that provides convenience for Users in managing goods. With this application, Users can simplify borrowing, transferring, and checking goods. The implementation of the Agile method can be a useful approach in designing the Mobile-based Indramayu State Polytechnic Inventory Application, because it can produce applications that are more in line with the needs of the Indramayu State Polytechnic agency, and can improve the management of goods at the Indramayu State Polytechnic Agency and faster and more efficient for Application Users.*

***Keywords*** *:Indramayu State Polytechnic, Inventory system, Mobile.*

MOTTO

“Bener jare wong tua. Ntas mangan kuh wareg”

KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah AWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan Skripsi dengan judul “Rancang Bangun Sistem *Inventory* Barang Pada Polindra berbasis *Website* menggunakan metode *Agile*”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Rekayasa Perangkat Lunak, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan Skripsi ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. *Allah subhanahu wa ta'ala*
2. Bapak Rofan Aziz, S.T., M.T., selaku Direktur Politeknik Negeri Indramayu (Polindra)
3. Bapak Eka Ismantohadi, S.Kom., M.Eng., selaku Ketua Jurusan Teknik Informatika
4. Bapak Dr. Mohammad Yani, S.T., M.T., M.Sc, selaku Dosen Pengampu Mata Kuliah Metodologi Penelitian yang telah memberikan ilmu dan pengetahuan akan kaidah penulisan dalam skripsi ini
5. Bapak Iryanto, S.Si., M.Si., selaku Dosen Pembimbing yang telah banyak memberikan bimbingan dan arahan dalam penulisan skripsi ini
6. Seluruh Bapak dan Ibu Dosen dan Staf Prodi Rekayasa Perangkat Lunak yang telah banyak membantu penulis selama masa perkuliahan
7. Orang Tua tercinta, terimakasih yang tak terhingga atas doa, semangat, kasih sayang, pengorbanan, dan ketulusannya dalam mendampingi penulis
8. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan skripsi dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan Skripsi ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan Skripsi ini. Semoga Skripsi ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terimakasih.

Indramayu, Agustus 2023

Penulis

DAFTAR ISI

Halaman

[HALAMAN PENGESAHAN iii](#_Toc144818377)

[PERNYATAAN KEASLIAN iv](#_Toc144818378)

[ABSTRAK v](#_Toc144818379)

[ABSTRACT vi](#_Toc144818380)

[MOTTO vii](#_Toc144818381)

[KATA PENGANTAR viii](#_Toc144818382)

[DAFTAR ISI x](#_Toc144818383)

[DAFTAR GAMBAR xiii](#_Toc144818384)

[DAFTAR TABEL xv](#_Toc144818385)

[DAFTAR LAMPIRAN xvi](#_Toc144818386)

[BAB I PENDAHULUAN 1](#_Toc144818387)

[1.1. Latar Belakang 1](#_Toc144818388)

[1.2. Rumusan Masalah 2](#_Toc144818389)

[1.3. Batasan Masalah 2](#_Toc144818390)

[1.4. Tujuan Penelitian 3](#_Toc144818391)

[1.5. Manfaat Penelitian 3](#_Toc144818392)

[1.6. Sistematika Penulisan 3](#_Toc144818393)

[BAB II LANDASAN TEORI 5](#_Toc144818394)

[2.1. Studi Literatur 5](#_Toc144818395)

[2.2. Metode *Agile* 6](#_Toc144818396)

[2.3. *Inventory* 6](#_Toc144818397)

[2.4. *Android* 7](#_Toc144818398)

[2.5. *Dart* 7](#_Toc144818399)

[2.6. *Flutter* 8](#_Toc144818400)

[2.7. *API* (*Application Programming Interface*) 8](#_Toc144818401)

[2.8. *Mysql* 9](#_Toc144818402)

[2.9. Visual Sudio Code 10](#_Toc144818403)

[2.10. Flowchart 11](#_Toc144818404)

[2.11. *Unifiled Modeling Language* (UML) 13](#_Toc144818405)

[2.12.1. Use case Diagram 14](#_Toc144818406)

[2.12.2. Activity diagram 14](#_Toc144818407)

[2.12.3. Sequence diagram 15](#_Toc144818408)

[2.12.4. Class Diagram 16](#_Toc144818409)

[2.12.5. Entity Relationship Diagram 17](#_Toc144818410)

[2.12. *Kuesioner* 17](#_Toc144818411)

[BAB III METODE PENELITIAN 19](#_Toc144818412)

[3.1. Metode Penelitian Perangkat Lunak 19](#_Toc144818413)

[3.2. Waktu dan Tempat Penelitian 19](#_Toc144818414)

[3.3. Metode Pengumpulan Data 19](#_Toc144818415)

[3.3.1. Wawancara 19](#_Toc144818416)

[3.3.2. Studi Literatur 19](#_Toc144818417)

[3.4. Tahapan Pelaksanaan 20](#_Toc144818418)

[3.4.1. Pengumpulan Data 20](#_Toc144818419)

[3.4.2. Membuat *Backlog* Produk 20](#_Toc144818420)

[3.4.3. Rapat Persiapan 21](#_Toc144818421)

[3.4.4. Sprint 21](#_Toc144818422)

[3.4.5. Implementasi Produk 21](#_Toc144818423)

[3.4.6. Penyusunan Laporan 22](#_Toc144818424)

[3.5. Rancangan Sistem 22](#_Toc144818425)

[3.5.1. Use Case Diagram 22](#_Toc144818426)

[3.5.2. Flowchart 23](#_Toc144818427)

[3.5.3. Activity Diagram 24](#_Toc144818428)

[3.5.4. Class Diagram 25](#_Toc144818429)

[3.5.5. Entity Relationship Diagram (ERD) 25](#_Toc144818430)

[3.5.6. Sequence Diagram 26](#_Toc144818431)

[3.6. Rencana Kegiatan 27](#_Toc144818432)

[BAB IV HASIL DAN PEMBAHASAN 28](#_Toc144818433)

[4.1. Hasil Penelitian 28](#_Toc144818434)

[4.2. Data Hasil Wawancara 28](#_Toc144818435)

[4.3. Fitur Aplikasi yang Dikembangkan Berdasarkan *User Story* 29](#_Toc144818436)

[4.4. Metode *Agile Scrum* 31](#_Toc144818437)

[4.5. Cakupan Yang Dikerjakan 34](#_Toc144818438)

[4.6. Artefak *Scrum* 35](#_Toc144818439)

[4.6.1. *Product Backlog* 35](#_Toc144818440)

[4.6.2. *Sprint Backlog* 38](#_Toc144818441)

[4.6.3. *Product Increment* 38](#_Toc144818442)

[4.7. Pelaksanaan *Sprint* 38](#_Toc144818443)

[4.7.1. *Sprint* 1 38](#_Toc144818444)

[4.7.2. *Sprint* 2 45](#_Toc144818445)

[4.7.3. *Sprint* 3 51](#_Toc144818446)

[4.7.4. *Sprint* 4 57](#_Toc144818447)

[4.7.5. *Sprint* 5 62](#_Toc144818449)

[4.8. Kuesioner Pengujian 68](#_Toc144818451)

[4.8.1. Pertanyaan Kuesioner 68](#_Toc144818452)

[4.8.2. Hasil Perhitungan Kuesioner 69](#_Toc144818453)

[4.8.3. Rekapitulasi Hasil Perhitungan Kuesioner 72](#_Toc144818454)

[4.8.4. Kesimpulan Pengujian 72](#_Toc144818455)

[BAB V PENUTUP 73](#_Toc144818456)

[5.1. Kesimpulan 73](#_Toc144818457)

[5.2. Saran 73](#_Toc144818458)

[DAFTAR PUSTAKA 74](#_Toc144818459)

[LAMPIRAN-LAMPIRAN 76](#_Toc144818460)

DAFTAR GAMBAR

Halaman

[Gambar 2. 1 Versi SDK *Android* 7](#_Toc144814390)

[Gambar 2. 2 Versi *Dart* 8](#_Toc144814391)

[Gambar 2. 3 Versi *Flutter* 8](#_Toc144814392)

[Gambar 2. 4 *API* Login 9](#_Toc144814393)

[Gambar 2. 5 Versi *Mysql* 10](#_Toc144814394)

[Gambar 2. 6 Tampilan awal *Visual Studio Code* 10](#_Toc144814395)

[Gambar 3. 1 Flowchart *Tahapan Pelaksanaan* 20](#_Toc144814376)

[Gambar 3. 2 Proses Metode Agile 21](#_Toc144814377)

[Gambar 3. 3 *Use Case* 22](#_Toc144814378)

[Gambar 3. 4 *Flowchart* sistem 23](#_Toc144814379)

[Gambar 3. 5 *Activity* diagrammobilitas 24](#_Toc144814380)

[Gambar 3. 6 *Activity* diagram peminjaman 24](#_Toc144814381)

[Gambar 3. 7 *Class Diagram* 25](#_Toc144814382)

[Gambar 3. 8 *ERD sistem* 26](#_Toc144814383)

[Gambar 3. 9 *Sequence diagram mobilitas* 26](#_Toc144814384)

[Gambar 3. 10 *Sequence* diagram peminjaman 27](#_Toc144814385)

[Gambar 4. 1 Sesi Wawancara dengan Petugas Bidang *Inventory* 28](#_Toc144813314)

[Gambar 4. 2 Komposisi Tim Scrum 31](#_Toc144813315)

[Gambar 4. 3 Ilustrasi Integrasi Database dengan *Mobile* App 34](#_Toc144813316)

[Gambar 4. 4 *Flowchart login* 40](#_Toc144813317)

[Gambar 4. 5 *Login* pengguna 40](#_Toc144813318)

[Gambar 4. 6 *Flowchart dashboard* 41](#_Toc144813319)

[Gambar 4. 7 Halaman *Dashboard* 41](#_Toc144813320)

[Gambar 4. 8 *Flowchart Scan Qr* 42](#_Toc144813321)

[Gambar 4. 9 Fitur Scan QR 42](#_Toc144813322)

[Gambar 4. 10 Dokumentasi Kegiatan 45](#_Toc144813323)

[Gambar 4. 11 *Flowchart* keranjang mobilitas 46](#_Toc144813324)

[Gambar 4. 12 Fitur keranjang mobilitas 47](#_Toc144813325)

[Gambar 4. 13 Flowchart keranjang mobilitas 47](#_Toc144813326)

[Gambar 4. 14 Fitur mobilitas 48](#_Toc144813327)

[Gambar 4. 15 *Flowchart* riwayat mobilitas 48](#_Toc144813328)

[Gambar 4. 16 Fitur riwayat mobilitas 49](#_Toc144813329)

[Gambar 4. 16 Dokumentasi Kegiatan 51](#_Toc144813330)

[Gambar 4. 18 *Flowchart* keranjang peminjaman 53](#_Toc144813331)

[Gambar 4. 19 Fitur Keranjang Peminjaman 53](#_Toc144813332)

[Gambar 4. 20 *Flowchart* Fitur peminjaman 54](#_Toc144813333)

[Gambar 4. 21 Fitur peminjaman 54](#_Toc144813334)

[Gambar 4. 22 Dokumentasi Kegiatan 56](#_Toc144813335)

[Gambar 4. 23 *Flowchart* proses persetujuan 58](#_Toc144813336)

[Gambar 4. 24 Fitur proses persetujuan 59](#_Toc144813337)

[Gambar 4. 25 *Flowchart* prose peminjaman 59](#_Toc144813338)

[Gambar 4. 26 Proses peminjaman 60](#_Toc144813339)

[Gambar 4. 27 Dokumentasi Kegiatan 62](#_Toc144813340)

[Gambar 4. 28 *Flowchart* persetujuan 63](#_Toc144813341)

[Gambar 4. 29 Fitur persetujuan pembimbing / penanggung jawab 64](#_Toc144813342)

[Gambar 4. 30 *Flowchart* persetujuan 64](#_Toc144813343)

[Gambar 4. 31 persetujuan wakil direktur 2 65](#_Toc144813344)

[Gambar 4. 32 *Flowchart* riwayat peminjaman 65](#_Toc144813345)

[Gambar 4. 33 Fitur riwayat pengguna 66](#_Toc144813346)

[Gambar 4. 34 Dokumentasi Kegiatan 68](#_Toc144813347)

DAFTAR TABEL

Halaman

[Tabel 2. 1 Simbol Arus 11](#_Toc144815066)

[Tabel 2. 2 Simbol Proses 12](#_Toc144815067)

[Tabel 2. 3 Simbol I/O 13](#_Toc144815068)

[Tabel 2. 4 Simbol *Use Case Diagram* 14](#_Toc144815069)

[Tabel 2. 5 Simbol *Activity Diagram* 15](#_Toc144815070)

[Tabel 2. 6 Simbol *Sequence Diagram* 15](#_Toc144815071)

[Tabel 2. 7 Simbol *Class Diagram* 16](#_Toc144815072)

[Tabel 2. 8 Simbol ERD 17](#_Toc144815073)

[Tabel 3.1 Nama *Use Case* 23](#_Toc144815674)

[Tabel 3. 2 Tabel Rencana Kegiatan 27](#_Toc144815675)

[Tabel 4. 1 Fitur yang dikembangkan 29](#_Toc144815693)

[Tabel 4. 3 *Product Backlog* 35](#_Toc144815694)

[Tabel 4. 4 *Sprint* *Backlog* pada *Sprint* 1 39](#_Toc144815695)

[Tabel 4. 5 Hasil pengerjaan *Sprint* *Backlog* 39](#_Toc144815696)

[Tabel 4. 6 Tabel Pengujian *Sprint* 1 43](#_Toc144815697)

[Tabel 4. 7 *Sprint* *Backlog* pada *Sprint* 2 45](#_Toc144815698)

[Tabel 4. 8 Hasil pengerjaan *Sprint* *Backlog* 46](#_Toc144815699)

[Tabel 4. 9 Tabel Pengujian *Sprint* 2 49](#_Toc144815700)

[Tabel 4. 10 *Sprint* *Backlog* pada *Sprint* 2 52](#_Toc144815701)

[Tabel 4. 11 Hasil pengerjaan *Sprint* *Backlog* 52](#_Toc144815702)

[Tabel 4. 12 Tabel Pengujian *Login* 55](#_Toc144815703)

[Tabel 4. 13 *Sprint* *Backlog* pada *Sprint* 2 57](#_Toc144815704)

[Tabel 4. 14 Hasil pengerjaan *Sprint* *Backlog* 58](#_Toc144815705)

[Tabel 4. 15 Tabel Pengujian *Sprint* 4 60](#_Toc144815706)

[Tabel 4. 16 *Sprint* *Backlog* pada *Sprint* 2 62](#_Toc144815707)

[Tabel 4. 17 Hasil pengerjaan *Sprint* *Backlog* 63](#_Toc144815708)

[Tabel 4. 18 Tabel Pengujian *Login* 66](#_Toc144815709)

[Tabel 4. 19 Penilaian kuesioner 68](#_Toc144815710)

[Tabel 4. 20 Pertanyaan kuesioner 69](#_Toc144815711)

[Tabel 4. 21 Hasil perhitungan kuesioner 69](#_Toc144815712)

DAFTAR LAMPIRAN

Lampiran 1 Laporan *Sprint*

Lampiran 2 Testing

Lampiran 3 Biodata Mahasiswa

BAB I   
PENDAHULUAN

* 1. Latar Belakang

*Inventory* barang didalam suatu lembaga menjadi hal yang penting bagi suatu lembaga pendidikan karena dari *Inventory* tersebut dapat mencatat *Inventory* barang yang dimiliki lembaga pendidikan tersebut. Oleh karena itu pihak lembaga pendidikan tersebut harus dapat mengelola *Inventory* barang dengan efektif dan efisien agar sesuai dengan tujuan dan berjalan baik. Politeknik Negeri Indramayu merupakan sebuah lembaga pendidikan pada era modern, yang harus dituntut untuk memberikan pelayanan, pendidikan, dan kemandirian bagi para mahasiswanya. Tak terlepas dari itu, lebaga pendidikan ini juga terus berusaha mengembangkan metode pembelajaran dan sistematika dalam mengurus dan mengelola berbagai sistemnya. Pada proses pengolahan data dan peminjaman sistem *Inventory* di Politeknik Negeri Indramayu yang masih belum menggunakan sistem yang terkomputerisasi dalam pendataan stok dan peminjaman *Inventory* masih dilakukan secara manual dengan mencatat disebuah buku tulis ataupun laporan sementara.

Bedasarkan latar belakang di atas, maka peneliti memberikan solusi yaitu dengan membuat sistem *Inventory* pada Politeknik Indramayu agar lebih akurat, mengelola data, penerimaan data, data peminjaman dan membuat laporan sistem *Inventory* barang yang terkomputerisasi pada Politeknik Negeri Indramayu. Tujuan penelitian ini berdasarkan permasalahan diatas yaitu mempermudah civitas academica Politeknik Negeri Indramayu untuk mengetahui data barang *Inventory* yang dimiliki, dapat memberikan informasi data *Inventory* barang yang lebih akurat, dapat memberikan kemudahan peminjaman *Inventory* barang yang ada di Politeknik Negeri Indramayu, dan menghasilkan report barang yang masuk, keluar, dan yang dipinjam secara terperinci. Penelitian ini diharapkan memberi manfaat bagi civitas academica Politeknik Negeri Indramayu sebagai bahan pertimbangan dalam menyelesaikan permasalahan yang dihadapi dalam pengolahan data dan peminjaman *Inventory* pada Politeknik Negeri Indramayu.

Aplikasi sistem invertori pada Politeknik Negeri Indramayu mempunyai dua jenis aplikasi yang berbeda yaitu aplikasi berbasis *Mobile* dan website. Aplikasi berbasis *Mobile* digunakan untuk melakukan beberapa aktivitas utama yaitu peminjaman *Inventory* dan distribusi *Inventory*. Sementara aplikasi berbasis website digunakan oleh admin dalam mengelola seluruh data yang dibutuhkan untuk operasional aplikasi secara keseluruhan (Maulana, 2023).

Dalam mengembangkan aplikasi sistem *Inventory* pada Politeknik Negeri Indramayu dibutuhkan metode pengembangan prangkat lunak secara fleksibel dan dapat menyelesaikan aplikasi dengan lebih cepat, andal, dan efektif. Hal tersebut diperlukan agar aplikasi yang dikembangkan dapat segera digunakan (Yusril et al., 2021). Oleh karena itu, penulis akan menggunakan metode pengembangan prangkat lunak dengan metode *Agile*, yang dapat fleksibel dalam segala perubahan berdasarkan kebutuhan.

* 1. Rumusan Masalah

Berdasarkan latar belakang yang dijabarkan, maka diperoleh suatu rumusan permasalahan yang menjadi dasar penelitian ini, yakni sebagai berikut :

1. Apa saja tantangan yang dihadapi dalam pengelolaan data dan peminjaman Inventory barang secara manual di Politeknik Negeri Indramayu?
2. Bagaimana pengembangan dan implementasi sistem Inventory berbasis mobile di Politeknik Negeri Indramayu dapat membantu mengatasi tantangan pengelolaan Inventory?
   1. Batasan Masalah

Untuk memfokuskan pembahasan, dapat diperoleh beberapa batasan masalah, di antaranya:

* + 1. Aplikasi Sistem *Inventory* dikembangkan dengan menggunakan *Framework* aplikasi *Mobile* *Flutter* versi 3.7.3 yang ditulis dengan menggunakan bahasa pemrograman *Dart* versi 2.20.1.
    2. Dalam mengintegrasikan aplikasi *Mobile* dengan *server*, penulis menggunakan API yang dibuat dengan *framework* *Laravel* versi 10.0.3 dan ditulis dengan bahasa pemrograman php versi 8.2.0.
    3. Sistem manajemen *database* yang digunakan merupakan Mysql.
    4. Aplikasi sistem *Inventory* diterapkan pada Politeknik Negeri Indramayu.
  1. Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini diantaranya :

1. Mengidentifikasi dengan rinci tantangan-tantangan yang terkait dengan pengelolaan data dan peminjaman Inventory barang secara manual di Politeknik Negeri Indramayu.
2. Merancang dan mengembangkan aplikasi Inventory berbasis mobile yang efisien, user-friendly, dan kemudahan akses bagi civitas Politeknik Negeri Indramayu.

Dengan mencapai tujuan-tujuan tersebut, penelitian ini diharapkan akan memberikan wawasan yang mendalam dan solusi yang berarti terkait efektivitas pengelolaan *Inventory* serta pengembangan sistem *Inventory* terkomputerisasi di lingkungan Politeknik Negeri Indramayu.

* 1. Manfaat Penelitian

Manfaat dari rancang bangun aplikasi sistem *Inventory* barang pada Politeknik Negeri Indramayu ini adalah sebagai berikut:

1. Penelitian ini akan membantu Politeknik Negeri Indramayu dalam meningkatkan efektivitas pengelolaan Inventory barang, dengan kemampuan mengecek, mendistribusi, dan meminjam barang secara lebih mudah.
2. Civitas academica Politeknik Negeri Indramayu akan lebih mudah dalam mengakses dan menggunakan informasi mengenai barang Inventory secara efektif.
   1. Sistematika Penulisan

Untuk memahami lebih jelas laporan ini, maka materi-materi yang tertera pada laporan skripsi ini dikelompokkan menjadi beberapa bab dengan sistematika penyampaian sebagai berikut:

**BAB I** : Pendahuluan, dalam hal ini penulis menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan skripsi.

**BAB II** : Landasan Teori, yaitu bab yang menguraikan tentang kajian pustaka baik dari buku-buku ilmiah, maupun sumber-sumber lain yang mendukung penelitian ini.

**BAB III** : Metode Penelitian, yaitu bab yang menguraikan tentang metode pelaksanaan, analisis kebutuhan sistem, perancangan sistem, *Flowchart*, perancangan database, dan perancangan *User interface*.

**BAB IV** : Hasil dan Pembahasan, yaitu bab yang menguraikan tentang tahapan analisis, desain, implementasi desain, hasil pengujian serta implementasinya yang berupa penjelasan teoritik, baik secara kualitatif, kuantitatif, maupun secara statik yang dibuat dalam pengujian program untuk memastikan kesesuaian program yang diharapkan penulis.

**BAB V** : Penutup, dalam hal ini penulis menguraikan tentang kesimpulan dan saran terhadap penelitian dengan berupa pendapat untuk melaksanakan sesatu yang belum diselesaikan secara maksimal.

BAB II   
LANDASAN TEORI

1. Studi Literatur

Sebelumnya terdapat penelitian mengenai Aplikasi Sistem *Inventory* Barang. Sehingga dalam penelitian ini, penulis berupaya untuk melakukan studi literatur sebagai salah satu alat dari penerapan metode penelitian. Beberapa studi literatur tersebut adalah:

1. Penelitian ini dilakukan oleh Edy Listiyan dan Egia Rosi Subhiyakto pada tahun 2021 yang berjudul “Rancang Bangun Sistem *Inventory* Gudang Menggunakan Metode Waterfall Studi Kasus Di Cv. Aqualux Duspha Abadi Kudus Jawa Tengah”. Penelitian ini yaitu membangun sebuah sistem *Inventory* barang pada Cv. Aqualux Duspha dalam monitoring keluar masuknya barang, persediaan barang dan alat yang tersimpan di gudang pusat. Yang dimana sebelumnya CV. Aqualux Duspha Abadi ini melakukan monitoring ketersediaan barang di gudang menggunakan data diatas kertas yang sering sekali terjadi ketidak sinkronan antara jumlah barang dan data yang ada (Listiyan & Subhiyakto, 2021).
2. Penelitian ini dilakukan oleh Muhamad Ihpaz Ramadhan dan Bullion Dragon Andah pada tahun 2018 dengan judul Implementasi “Sistem Informasi Peminjaman Aset kampus Berbasis Web Pada Direktorat Administrasi umum Universitas Budi Luhur Dengan Metodologi berorientasi Obyek” Penelitian ini bertujuan untuk mempermudah proses peminjaman karena berbasis web, sehingga informasi yang dibutuhkan untuk memberikan sanksi dan menyiapkan barang yang akan dipinjam dapat diakses dengan lebih praktis. Penelitian ini telah menghasilkan sebuah sistem peminjaman aset yang berbasis web, menggunakan metode pemrograman berorientasi objek (object-oriented programming). Sistem ini dapat menghasilkan berbagai laporan yang berguna, seperti laporan permintaan peminjaman aset per periode, laporan permintaan peminjaman ruang per periode, laporan barang yang belum dikembalikan, laporan perilaku buruk pengguna, serta laporan barang yang paling sering dipinjam. Sistem Peminjaman Aset ini dibuat menggunakan bahasa pemrograman *PHP* dengan *framework* *Codeigniter* dan database *Mysql*. Dengan sistem ini, diharapkan proses peminjaman aset menjadi lebih efisien dan teratur, serta memungkinkan pengelolaan data dan laporan secara lebih efektif (Ramadhan & Andah, 2018).
3. Metode *Agile*

Metode *Agile* adalah sebuah pendekatan dalam pengembangan perangkat lunak yang berfokus pada proses kerja berulang yang terdiri dari aturan dan solusi yang telah disepakati. Pendekatan ini juga menekankan kolaborasi terstruktur dan terorganisir antar tim.

Metode *Agile* sangat sesuai untuk proyek-proyek jangka pendek karena kemampuannya dalam beradaptasi dengan mudah terhadap perubahan dalam suatu proyek. Dalam *Agile* *Development*, nilai yang paling penting diberikan pada tim yang dapat mengambil keputusan dengan cepat dan tepat.

Keputusan yang diambil tidak hanya cepat dalam segi waktu, tetapi juga memiliki kualitas prediksi yang baik. Sehingga keputusan tersebut mampu mengatasi permasalahan yang ada tanpa menimbulkan masalah baru.

Proses pengembangan *Agile* ini didasarkan pada konsep yang diusung oleh *Agile* Manifesto, yang dikembangkan oleh 14 tokoh penting dalam industri pengembangan perangkat lunak (binaracademy.com).

1. *Inventory*

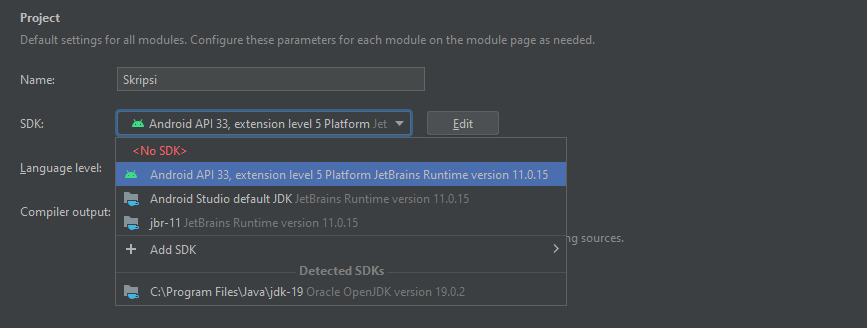
Sistem *Inventory* adalah suatu bagian dari proses perusahaan yang ditawarkan untuk diproduksi, maupun barang jadi yang disediakan untuk memenuhi permintaan dari konsumen sewaktu-waktu dan disimpan serta dipelihara menurut aturan tertentu dalam bentuk siap pakai dan disimpan dalam suatu basis data. *Inventory* juga dapat berupa barang-barang yang biasanya terdapat di gudang tertutup, lapangan, gudang terbuka, atau tempat penyimpanan lainnya, baik dalam bentuk bahan mentah, barang jadi, barang untuk keperluan operasional, maupun barang untuk keperluan proyek (Gosal & Rustam, 2022) .

Menurut sumber lain *Inventory* yaitu merupakan kumpulan sumber daya, baik berupa bahan mentah maupun barang jadi, yang disediakan oleh perusahaan untuk memenuhi permintaan dari konsumen. Definisi lain menyebutkan bahwa *Inventory* merupakan teknik manajemen material yang berhubungan dengan persediaan (Mufida et al., 2019).

1. *Android*

*Android* merupakan sebuah sistem operasi *Mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan platform terbuka yang memungkinkan para pengembang untuk membuat aplikasi mereka (Azis et al., 2020).

*Android* merupakan sistem operasi berbasis *Linux* yang digunakan pada perangkat bergerak seperti telepon seluler. Definisi lainnya menyebutkan bahwa *Android* adalah sistem operasi untuk smartphone dan tablet. Dalam era modern ini, teknologi semakin maju dan *Android* tidak hanya terbatas pada penggunaan di smartphone atau tablet, tetapi juga sudah dapat digunakan di komputer atau laptop. Sifat open source dari sistem Android memungkinkan para programmer untuk membuat dan memodifikasi aplikasi pada platform ini. Peran programmer sangat penting dalam pengembangan Android karena sistemnya bersifat *open* *source*. Pengguna *smartphone* dapat menikmati aplikasi yang dibuat oleh *programmer* secara gratis atau berbayar melalui *Play* *Store*. (Arsyad & Lestari, 2020). Berikut adalah versi SDK *Android* yang digunakan penulis dalam melakukan penelitian ini yang dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Versi SDK *Android*

1. *Dart*

Bahasa pemrograman *Dart* merupakan bahasa pemrograman yang telah dioptimalkan untuk pengembangan aplikasi klien di berbagai *platform*. *Dart* sangat cocok digunakan dalam pengembangan aplikasi klien karena bahasa ini memiliki fokus yang tinggi pada pengembangan. Salah satu contoh fitur *Dart* yang mengutamakan pengembangan aplikasi adalah fitur hot *reload*, yang memungkinkan pengembang melihat perubahan tampilan secara langsung saat sedang menulis kode, tanpa perlu menunggu proses recompile atau rebuild. *Dart* juga memungkinkan pengembangan aplikasi dengan mudah untuk *web*, *Mobile*, dan desktop (Taruna et al., 2022). Berikut adalah versi *Dart* yang digunakan penulis dalam melakukan penelitian ini yang dapat dilihat pada Gambar 2.2.

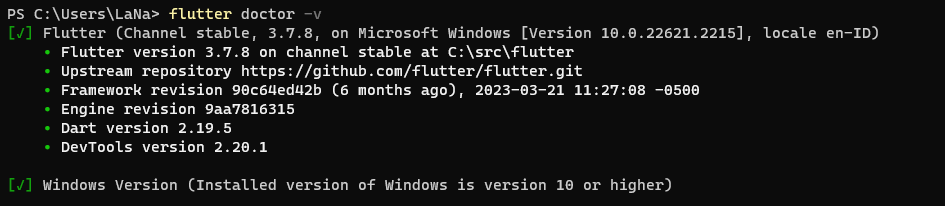


Gambar 2. 2 Versi *Dart*

1. *Flutter*

*Flutter* adalah toolkit UI portabel yang dikembangkan oleh Google, digunakan untuk membuat aplikasi yang dikompilasi secara native untuk berbagai platform seperti seluler atau *Mobile*, web, dan desktop. *Flutter* memungkinkan pengembang untuk menggunakan satu basis kode untuk mengembangkan aplikasi yang dapat berjalan di berbagai *platform* tersebut. *Flutter* telah digunakan oleh banyak pengembang dan organisasi di seluruh dunia, dan tersedia secara gratis untuk digunakan atau bersifat *open source*.

*Flutter* menggunakan bahasa pemrograman *Dart* karena *Dart* memenuhi keempat kriteria penilaian yang dilakukan oleh pengembang *Flutter*. Bahasa *Dart* dipilih karena kemampuannya untuk mengembangkan aplikasi dengan efisien dan performa yang tinggi di platform-platform yang didukung oleh *Flutter* (Taruna et al., 2022). Berikut adalah versi *Flutter* yang digunakan penulis dalam melakukan penelitian ini yang dapat dilihat pada Gambar 2.3.

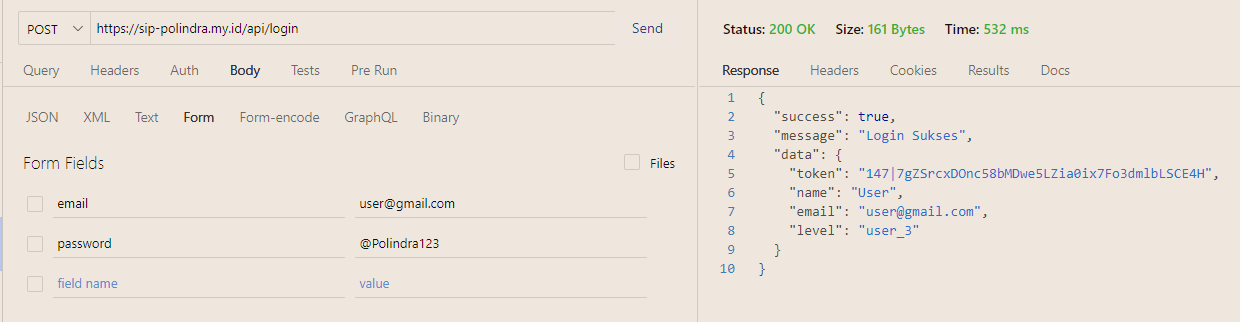


Gambar 2. 3 Versi *Flutter*

1. *API* (*Application Programming Interface*)

*API* adalah sebuah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari suatu program. Dengan menggunakan *API,* pengembang dapat menggunakan fungsi-fungsi yang sudah ada pada aplikasi lain tanpa perlu membuatnya dari awal. Pada konteks website, *API* merupakan panggilan fungsi melalui *Hyper Text Transfer Protocol* *(HTTP)* dan menerima respons dalam bentuk *Extensible Markup Language (XML)* atau *JavaScript Object Notation (JSON).*

Penggunaan *API* memiliki tujuan untuk berbagi data antara aplikasi yang berbeda. Selain itu, penggunaan *API* juga bertujuan untuk mempercepat proses pengembangan aplikasi dengan menyediakan fungsi-fungsi terpisah sehingga pengembang tidak perlu merancang fitur yang serupa secara mandiri*. API* yang bekerja pada tingkat sistem operasi membantu aplikasi dalam berkomunikasi dengan lapisan dasar dan satu sama lain, mengikuti serangkaian protokol dan spesifikasi yang telah ditentukan (Hasanuddin, Asgar, & Hartono, 2022). Berikut adalah salah satu *API* yang digunakan penulis dalam melakukan penelitian ini yang dapattt dilihat pada gambar 2.4.



Gambar 2. 4 *API* Login

1. *Mysql*

*Mysql* adalah sistem manajemen basis data relasional yang bersifat open source. *Mysql* mendukung pengelolaan database mulai dari yang berukuran kecil hingga yang sangat besar. *SQL* (*Structured Query Language*) adalah bahasa terstruktur yang dirancang terutama untuk pemrosesan basis data. *American National Standards Institute* (ANSI) mendefinisikan *SQL* untuk pertama kalinya pada tahun 1986. *SQL* juga dapat dipandang sebagai antarmuka standar untuk sistem manajemen basis data relasional, termasuk yang berjalan pada komputer pribadi. *SQL* memungkinkan pengguna untuk mengakses dan mengatur data dalam database. *SQL* lebih mudah digunakan daripada bahasa pemrograman, tetapi lebih sulit untuk dikuasai daripada spreadsheet dan alat pemrosesan data. *SQL* dapat digunakan untuk melakukan investigasi interaktif, membuat laporan ad hoc, atau disematkan dalam program aplikasi. *SQL* juga merupakan bahasa pemrograman yang dirancang khusus untuk mengirim perintah kueri ke database. Sebagian dari standar *SQL* didukung oleh setiap sistem manajemen basis data. Dengan demikian, *SQL* adalah kueri atau bahasa yang dibangun ke dalam sistem manajemen basis data seperti *Mysql*, *PostgreSQL*, *Oracle*, dan lainnya. *SQL* juga didukung oleh *database* berbasis *non-server* seperti MS Access dan Paradox. (Novendri, Saputra, & Firman, 2019). Berikut adalah versi *Mysql* yang digunakan penulis dalam melakukan penelitian ini yang dapat dilihat pada Gambar 2.5.

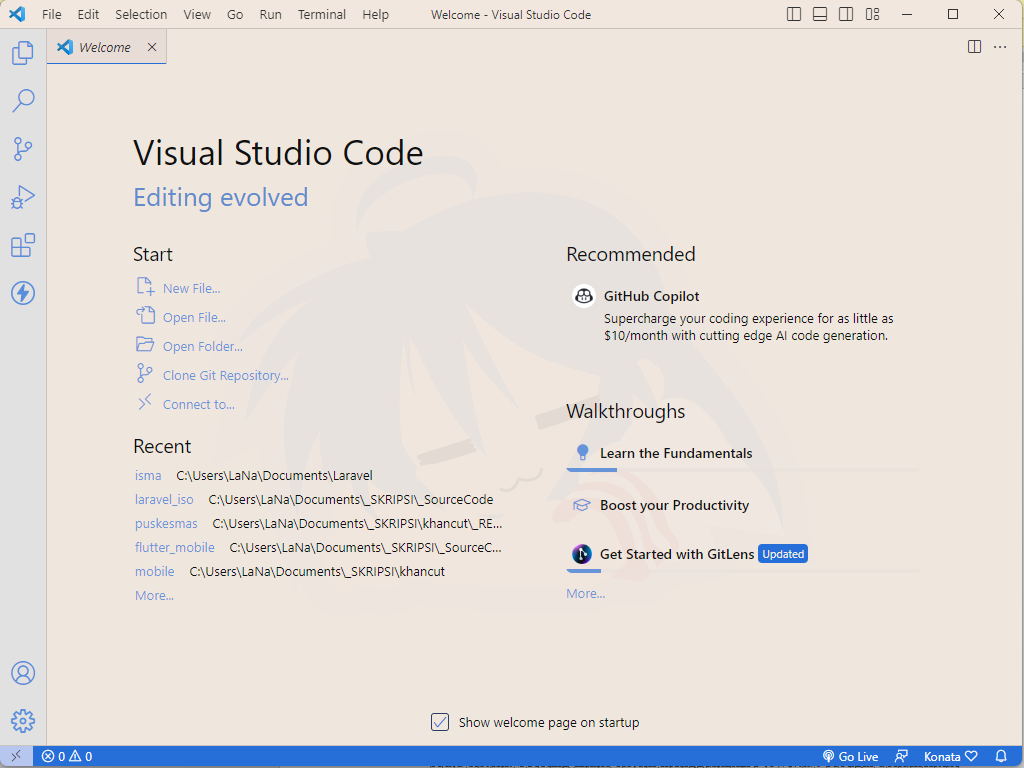


Gambar 2. 5 Versi *Mysql*

1. Visual Sudio Code

Visual Studio Code adalah sebuah perangkat lunak editor kode yang dikembangkan oleh Microsoft untuk berbagai sistem operasi seperti Linux, Windows, dan macOS. Perangkat ini dilengkapi dengan kontrol git yang terintegrasi dengan GitHub, sehingga memudahkan pengguna untuk melakukan penyesuaian dan memanfaatkan fungsionalitas tambahan.

Visual Studio Code juga menyediakan dokumentasi pengembangan serta bantuan untuk Node.js, Microsoft Script, ASP.NET 5, dan alat-alat lain yang membantu dalam membangun aplikasi Node.js. Meskipun masih dalam pengembangan aktif, Visual Studio Code dapat digunakan oleh siapa saja untuk membuat aplikasi web (Friandika Utamandani, 2021). Berikut adalah tampilan awal dari *Visual Studio Code* yang dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Tampilan awal *Visual Studio Code*

1. Flowchart

*Flowchart*, yang juga dikenal sebagai diagram alur, adalah jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam suatu sistem. Seorang analis sistem menggunakan *Flowchart* sebagai dokumentasi yang memberikan bukti untuk menjelaskan gambaran logis sistem yang akan dibangun kepada programmer. *Flowchart* digambarkan menggunakan simbol-simbol, dimana setiap simbol mewakili proses tertentu. *Flowchart* membantu menggambarkan urutan proses dengan lebih jelas. Selain itu, dengan menggunakan *Flowchart*, penambahan proses baru dapat dilakukan dengan mudah. Setelah *Flowchart* selesai dibuat, tugas programmer adalah menerjemahkan desain logis tersebut menjadi program yang sesuai dengan bahasa pemrograman yang telah disepakati (Rosaly & Prasetyo, 2019).

Untuk mengetahui simbol-simbol standar yang umum digunakan dalam proses pembuatan *flowchart*. Berikut adalah simbol-simbol *flowchart* yang dibagi kedalam 3 kategori (Rosaly, R., & Prasetyo, A., 2019) :

1. Simbol Arus (*Flow Direction Symbols)*,adalah simbol yang termasuk dalam kategori simbol penghubung. Beberapa simbol yang masuk kedalam kategori ini dapat dilihat pada tabel berikut ini :

Tabel 2.1 Simbol Arus

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Flow Direction Symbol / Connecting Line* | Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses |
|  | *Connector* | Digunakan untuk menyatakan sambungan dari proses satu ke proses berikutnya di halaman yang sama |
|  | *Offline Connector* | Digunakan untuk menyatakan sambungan dari proses satu ke proses berikutnya di halaman yang berbeda |

1. Simbol Proses *(Processing Symbols)*, adalah simbol yang digunakan untuk menyatakan simbol yang berkaitan dengan serangkaian proses yang dilakukan. Beberapa simbol yang masuk kedalam kategori ini dapat dilihat pada tabel berikut ini :

Tabel 2.2 Simbol Proses

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Processing* | Digunakan untuk menunjukkan pengolahan data yang akan dilakukan di dalam komputer |
|  | *Manual Operation* | Digunakan untuk menunjukkan pengolahan data yang tidak dilakukan oleh komputer |
|  | *Decision* | Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu |
|  | *Terminal* | Digunakan untuk memulai atau mengakhiri program |
|  | *Manual Input Symbol* | Digunakan untuk menginputkan data secara manual melalui *keyboard* |

1. Simbol I/O *(Input / Output)*, adalah simbol yang digunakan untuk menyatakan masukan dan keluaran. Simbol-simbol ini membantu dalam menunjukkan interaksi antara sistem dan lingkungannya. Beberapa simbol yang masuk kedalam kategori ini dapat dilihat pada tabel berikut ini:

Tabel 2.3 Simbol I/O

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Input / Output* | Digunakan untuk menyatakan *input* dan *output* tanpa melihat jenisnya |
|  | *Disk Storage* | Digunakan untuk menyatakan masukan dan keluaran yang berasal dari d*isk* |
|  | *Dokument* | Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen |
|  | *Display* | Digunakan untuk menyatakan keluaran melalui layer monitor |

1. *Unifiled Modeling Language* (UML)

UML merupakan sebuah bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Artefak tersebut dapat berupa model, deskripsi, atau perangkat lunak yang terlibat dalam proses pembuatan perangkat lunak. UML juga dapat digunakan dalam pemodelan bisnis dan sistem non-perangkat lunak lainnya. UML adalah bahasa pemodelan yang berbasis pada konsep orientasi objek. Pembuatan UML melibatkan Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah naungan Rational Software Corps. UML menyediakan notasi-notasi yang membantu dalam pemodelan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, tetapi juga digunakan dalam hampir semua bidang yang membutuhkan pemodelan. (Prasetya et al., 2022).

Ada beberapa UML yang digunakan dalam mendesain aplikasi ini, yaitu ada *Usecase* Diagram*, Flowchart* Diagram*, Activity* Diagram*, Sequence* Diagram*, Class* Diagram*,* dan *Entity Relationship* Diagram *(*ERD*).*

* + 1. Use case Diagram

*Use case* diagram adalah satu dari berbagai jenis diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya (Prasetya et al., 2022).

Tabel 2.4 Simbol *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Actor* | Menspesifikasikan himpunan pesan yang pengguna mainkan ketika berinteraksi dengan *use case* |
|  | *Include* | Relasi *use case* tambahan ke sebuah *use case*, dimana memerlukan *use case* ini untuk menjalankan fungsinya |
|  | *Extend* | Menspesifikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan |
|  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |
|  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas |
|  | *Use Case* | Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor* |

* + 1. Activity diagram

*Activity*, atau "diagram aktivitas" dalam bahasa Indonesia, adalah diagram yang dapat mensimulasikan proses yang terjadi dalam sebuah sistem. Gambar vertikal menunjukkan rangkaian proses suatu sistem. Alur atau aktivitas dapat berupa rangkaian menu atau proses bisnis yang ada dalam sistem (Prasetya et al., 2022).

Tabel 2.5 Simbol *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Start Point* | Penanda *workflow* mulai |
|  | *End Point* | Penanda *workflow* mulai |
|  | *Activites* | Menunjukan aktivitas yang dilakukan pada workflow |
|  | *Fork* (Percabangan) | Menunjukan adanya percabangan secara paralel dari aktivitas |
|  | *Decision* | Menunjukan suatu kondisi dimana ada kemungkinan perbedaan transisi |

* + 1. Sequence diagram

*Sequence* Diagram, juga dikenal sebagai diagram urutan, adalah sebuah diagram yang digunakan untuk menjelaskan dan menunjukkan interaksi antar objek dalam sebuah sistem secara menyeluruh. Diagram urutan juga menunjukkan pesan atau perintah yang dikirim, serta waktu pelaksanaannya. Biasanya, barang-barang yang berkaitan dengan proses operasi diurutkan dari kiri ke kanan. (Prasetya et al., 2022).

Tabel 2.6 Simbol *Sequence Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Actor* | Menspesifikasikan himpunan pesan yang pengguna mainkan ketika berinteraksi dengan *sequence* |
|  | *Lifeline* (Garis Hidup­­­­­) | Menyatakan kehidupan pada suatu objek yang ada |
|  | *Object* | Merupakan suatu interaksi objek dengan pesan |
|  | Pesan Tipe *Create* | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
|  | Pesan Tipe *Send* | Meyatakan bahwa suatu objek mengirimkan data informasi ke objek lainya |
|  | Pesan Tipe *Return* | Meyatakan bahwa suatu objek mengirimkan data hasil informasi ke objek lainya |

* + 1. Class Diagram

*Class Diagram* menggambarkan hubungan antara *class-class* di mana atribut dan fungsi pada suatu objek (Nazir dkk, 2022). Diagram ini statis karena *class diagram* menggambarkan hubungan apa yang terjadi ketika *class* terkait (Prasetya et al., 2022).

Tabel 2. 7 Simbol *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Class* | Pimpunan dari obyek-obyek yang berbagi atribut serta operasi yang sama. |
|  | *Inheritance/ Generalization* | Hubungan dimana obyek anak (descendent) berbagi perilaku dan struktur data dari obyek yang ada di atasnya obyek induk (ancestor). |
|  | *Composition* | Hubungan antara dua *class* di mana salah satu *class* merupakan bagian dari *class* lain, tetapi salah satu dari *class* tidak dapat berdiri sendiri tanpa *class* lainnya |
|  | *Aggregation* | Hubungan antara dua *class* di mana salah satu *class* merupakan bagian dari *class* lain, tetapi dua *class* ini dapat berdiri masing-masing |
|  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

* + 1. Entity Relationship Diagram

*ERD (Entity Relationship Diagram)* adalah representasi grafis dari data yang dimodelkan dalam bentuk diagram, yang didasarkan pada entitas-entitas dari dunia nyata dan hubungan antara entitas tersebut. *ERD* digunakan untuk menggambarkan secara sistematis hubungan antar entitas dalam suatu sistem database, dengan menggunakan simbol-simbol yang memudahkan pemahaman. *ERD* menggunakan notasi dan simbol tertentu untuk menggambarkan struktur dan hubungan antara data (WK, 2022).

Tabel 2.8 Simbol ERD

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | *Entity* | Bagian yang merupakan data inti yang akan di simpan pada *database* yang menjadi bakal tabel pada basis data |
|  | *Attribute* | Bagian yang menjadi *field* atau kolom yang digunakan pada suatu entitas. |
|  | *Key Attribute* | Field atau kolom yang di simpan pada suatu entitas digunakan sebagai kunci primer akses data yang diinginkan dan bersifat unik. |
|  | *Relationship* | Bagian yang menghubungkan antar entitas, biasanya diawali dengan kata kerja. |
|  | *Relation* | Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian maksimum keterhubungan antara entitas satu dengan entitas lainya. |

1. *Kuesioner*

*Kuesioner* atau angket merupakan metode pengumpulan informasi yang melibatkan penyampaian serangkaian pertanyaan atau pernyataan kepada individu yang berperan sebagai responden. Responden merespons kuesioner dengan menandai (X) pada opsi jawaban yang telah disediakan, yang biasanya terdiri dari lima pilihan yang berbeda. Setiap pertanyaan dinilai dengan skor berikut: (Mamesah, Rompas, & Katuuk, 2018)

1 = Sangat Tidak setuju (STS)

2 = Tidak setuju

3 = Kurang Setuju

4 = Setuju

5 = Sangat Setuju

BAB III  
METODE PENELITIAN

1. Metode Penelitian Perangkat Lunak

Dalam pembuatan aplikasi berbasis *Mobile* ini meliputi rumusan masalah, pengumpulan data, perancangan sistem, implementasi system dan pengujian system.

Metode penelitian sistem atau *software development lifecycle (SDLC)* yang digunakan dalam rancang bangun sistem ini adalah metode *Agile*.

1. Waktu dan Tempat Penelitian

Penelitian ini dilakukan di kampus Politeknik Negeri Indramayu khususnya dibagian yang bertanggung jawab dalam mengelola *Inventory* kampus, dengan durasi penelitian selama 4 bulan, mulai dari bulan April hingga Juli 2023. Untuk mendapatkan data-data pendukung, penulis menggunakan teknik pengumpulan data sebagai berikut:

1. Metode Pengumpulan Data

Dalam proses pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Pengumpulan data yang dilakukan oleh penulis yaitu menggunakn dua metode diantaranya:

1. Wawancara

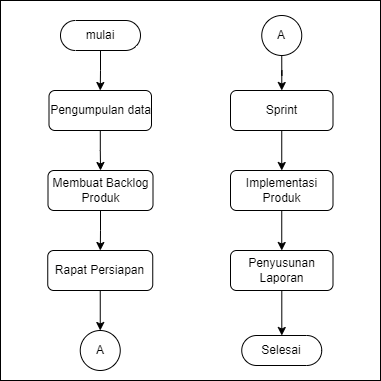
Penulis juga melakukan pengumpulan data dengan teknik wawancara dilakukan dengan menanyakan secara langsung ke narasumber yang bersangkutan yaitu pihak bagian *Inventory* Politeknik Negeri Indramayu. Pertanyaan-pertanyaan diajukan langsung kepada pihak *Inventory* selaku pemilik aplikasi dengan tujuan untuk mendapatkan keterangan atau data data penelitian yang diperlukan. Jawaban dari pihak *Inventory* ini dicatat atau direkam untuk kemudian dianalisis dalam penelitian ini.

1. Studi Literatur

Studi Literatur merupakan metode yang dilakukan oleh penulis untuk mengumpulkan data dengan melalui studi atau penelitian dokumen-dokumen atau sumber tertulis lainnya, termasuk buku, jurnal ilmiah, internet, dan majalah yang menjadi referensi dalam pembuatan aplikasi sesuai dengan tujuan penelitian ini. Metode ini digunakan untuk memperoleh data sekunder yang relevan dengan objek penelitian.

1. Tahapan Pelaksanaan

Pada penelitian ini, terdapat beberapa tahap yang telah disesuaikan dengan metode pengembangan perangkat lunak *Agile*. Untuk tahapan tersebut dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Flowchart *Tahapan Pelaksanaan*

Pada Gambar 3.1 terdapat 6 tahapan dalam pelaksanaan penelitian, setiap tahapan akan dibahas di bawah ini:

1. Pengumpulan Data

Pengumpulan data akan dilaksanakan dengan menggunakan teknik studi literatur, dan wawancara. Untuk setiap teknik pengumpulan data telah dijelaskan di atas.

1. Membuat *Backlog* Produk

*Blacklog* produk dibuat untuk membuat daftar tugas-tugas yang akan dikerjakan selama perancangan dan pembuatan sistem informasi aplikasi inventori Politeknik Negeri Indramayu. *Backlog* perlu di urutkan menurut skala prioritas, untuk mengetahui tugas yang paling penting dan harus diselesaikan terlebih dahulu. Pada setiap tugas juga perlu ditambahkan poin estimasi, untuk memperkirakan besar usaha dalam menyelesaikan setiap tugas. *Backlog* produk dapat berubah sewaktu-waktu selama pengembangan produk, karena setiap kebutuhan baru kemungkinan akan didapatkan pada saat pengembangan produk sedang berjalan.

1. Rapat Persiapan

Dengan adanya rapat ini, tim memiliki pemahaman yang jelas tentang apa yang harus dicapai selama Sprint, sehingga memungkinkan mereka untuk bekerja secara terfokus dan efisien.

1. Sprint

*Sprint* digunakan sebagai pengukuran rentang waktu tertentu untuk menyelesaikan beberapa tugas. Setiap *Sprint* terdapat beberapa aktifitas yaitu Perancangan *Sprint*, Pertemuan Harian, Pengembangan Produk, Ulasan *Sprint* atau *Demo* *Sprint*, dan Refreksi *Sprint*. Dalam pengembangan aplikasi SIP, penulis merencanakan sebanyak 12 *Sprint* untuk menyelesaikan seluruh *Backlog* produk. Setiap *Sprint* mempunyai waktu 1 minggu. Pada setiap *Sprint* terdapat proses *Requirement, Design*, *Develop*, *test, dan deploy*, untuk alur proses tersebut dapat dilihat pada Gambar 3.2:



Gambar 3. 2 Proses Metode Agile

1. Implementasi Produk

Implementasi produk digunakan sebagai upaya dalam mencari solusi atas permasalahan yang ada dan untuk mengidentifikasi kelemahan sistem agar sesuai dengan kebutuhan dari aplikasi Sistem *Inventory* barang ini, dengan tujuan memberikan kemudahan dalam proses keluar masuknya barang, melihat mobilitas barang dan juga proses dari pengelolaan *Inventory* barang. Bagi bidang *Inventory* kampus Politeknik Negeri Indramayu, aplikasi ini diharapkan dapat membantu mempermudah mengelelola barang serta memberikan pelayanan yang lebih baik untuk kampus Politeknik Negeri Indramayu.

1. Penyusunan Laporan

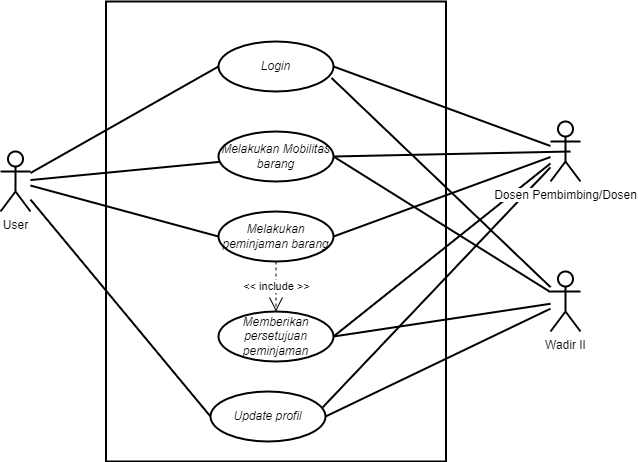
Setelah seluruh tahapan dilaksanakan, selanjutnya akan disusun laporan proses dan hasil dari yang telah penulis jalankan. Dengan dibuatnya laporan terkait dari skipsi ini, semoga dapat menjadi referensi atau panduan kepada peneliti selanjutnya yang akan menerapkan pengembangan perangkat lunak metode *Agile* dalam mengembangkan suatu perangkat lunak.

1. Rancangan Sistem

Pada tahap rancangan sistem ini di detailkan dengan bantuan tools UML (*Unified Modeling Language*) sebagai gambaran dari alur proses system yang digunakan penulis dalam menyusun skripsi ini dalam rancangan sistem dapat digambar dengan menggunakan:

1. Use Case Diagram

Pada perancangan usecase di dalam Aplikasi ini terdapat tiga aktor yang terlibat dalam sistem Inventori barang berbasis *mobile* pada Politeknik Negeri Indramayu yaitu diantaranya User, Dosen/Penanggung jawab dan Wakil Direktur II (Wadir II). *Usecase diagram* dari sistem ini dapat dilihat pada gambar 3.3.



*Profil*

Gambar 3. 3 *Use Case*

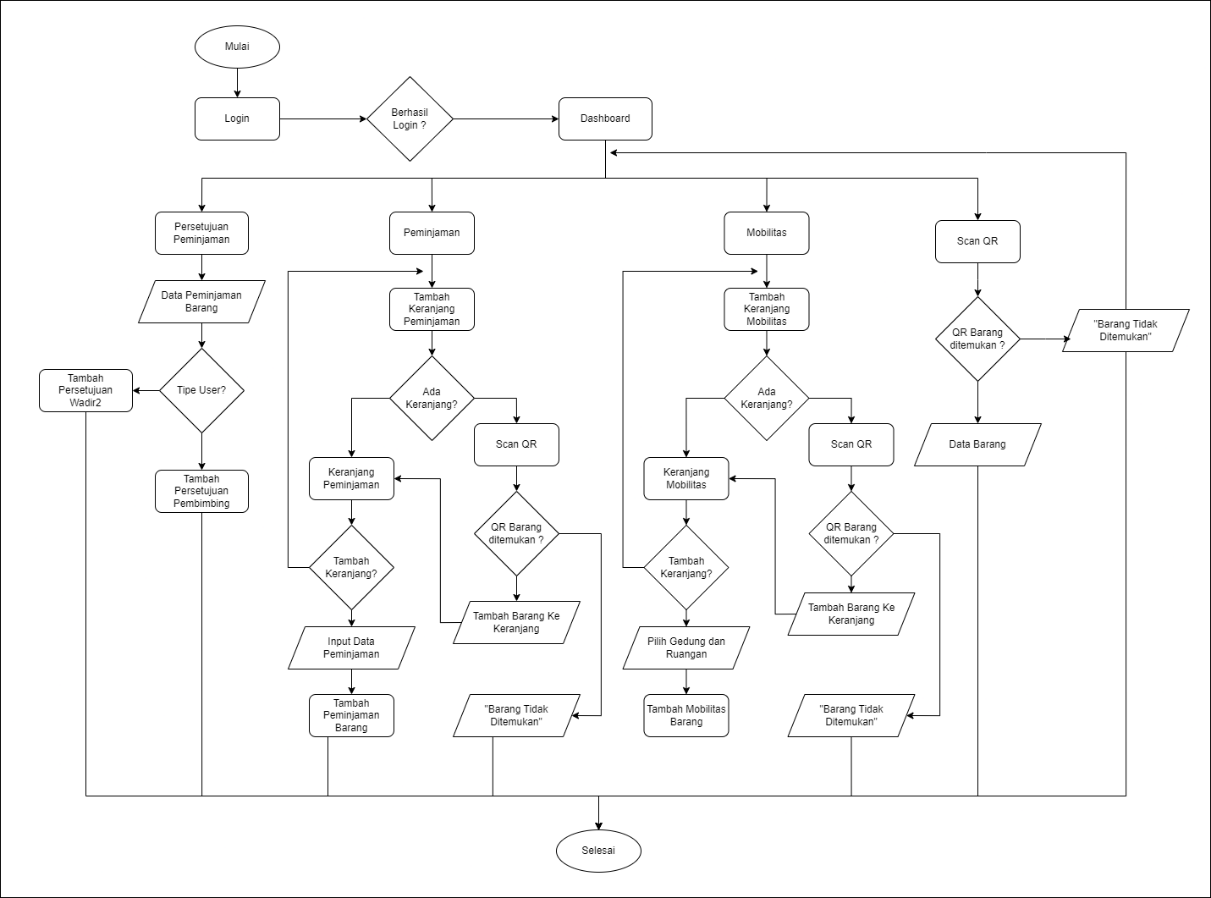
Aktor tersebut yang nantinya dapat mengelola sistem dapat dilihat pada Tabel 3. berikut:

Tabel 3.1 Nama *Use Case*

|  |  |  |  |
| --- | --- | --- | --- |
| Aktor | User (Mahasiswa): | Dosen/Penanggung jawab | Wakil Direktur II (Wadir II) |
| Nama  *Use Case* | 1) login, | 1) login, | 1) login, |
| 2) mobilitas, | 2) mobilitas, | 2) mobilitas, |
| 3) peminjaman, | 3) peminjaman, | 3) peminjaman, |
| 4) melihat profil, | 4) persetujuan, | 4) persetujuan, |
|  | 5) melihat profil, | 5) melihat profil, |

1. Flowchart

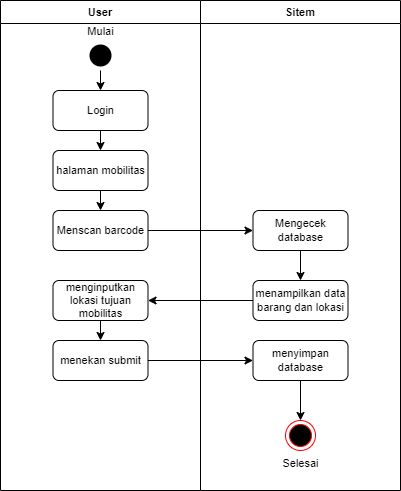
*Flowchart* sistem digunakan untuk merencanakan dan mendokumentasikan proses kerja dari sistem secara keselurahan. Berikut *flowchart* sistem dapat dilihat pada Gambar 3.4



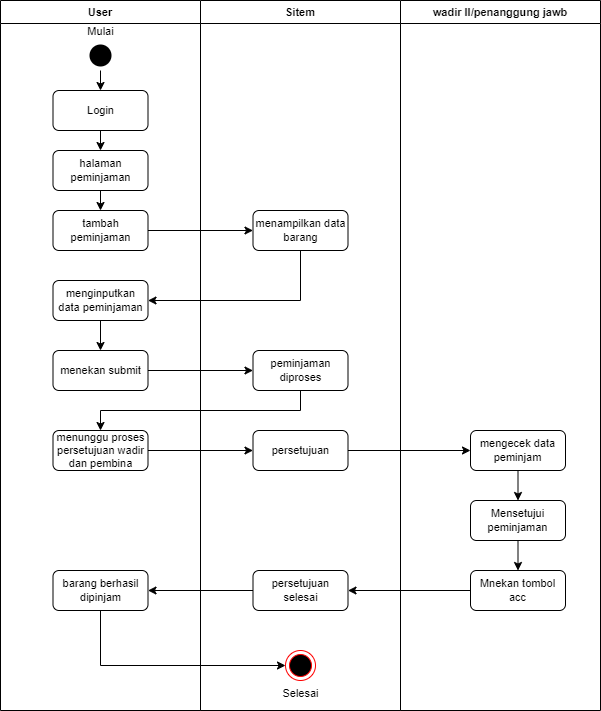
Gambar 3. 4 *Flowchart* sistem

1. Activity Diagram

Pada perancangan aplikasi terdapat *Activity diagram* dari penelitian ini yang dapat dilihat pada gambar dibawah ini.



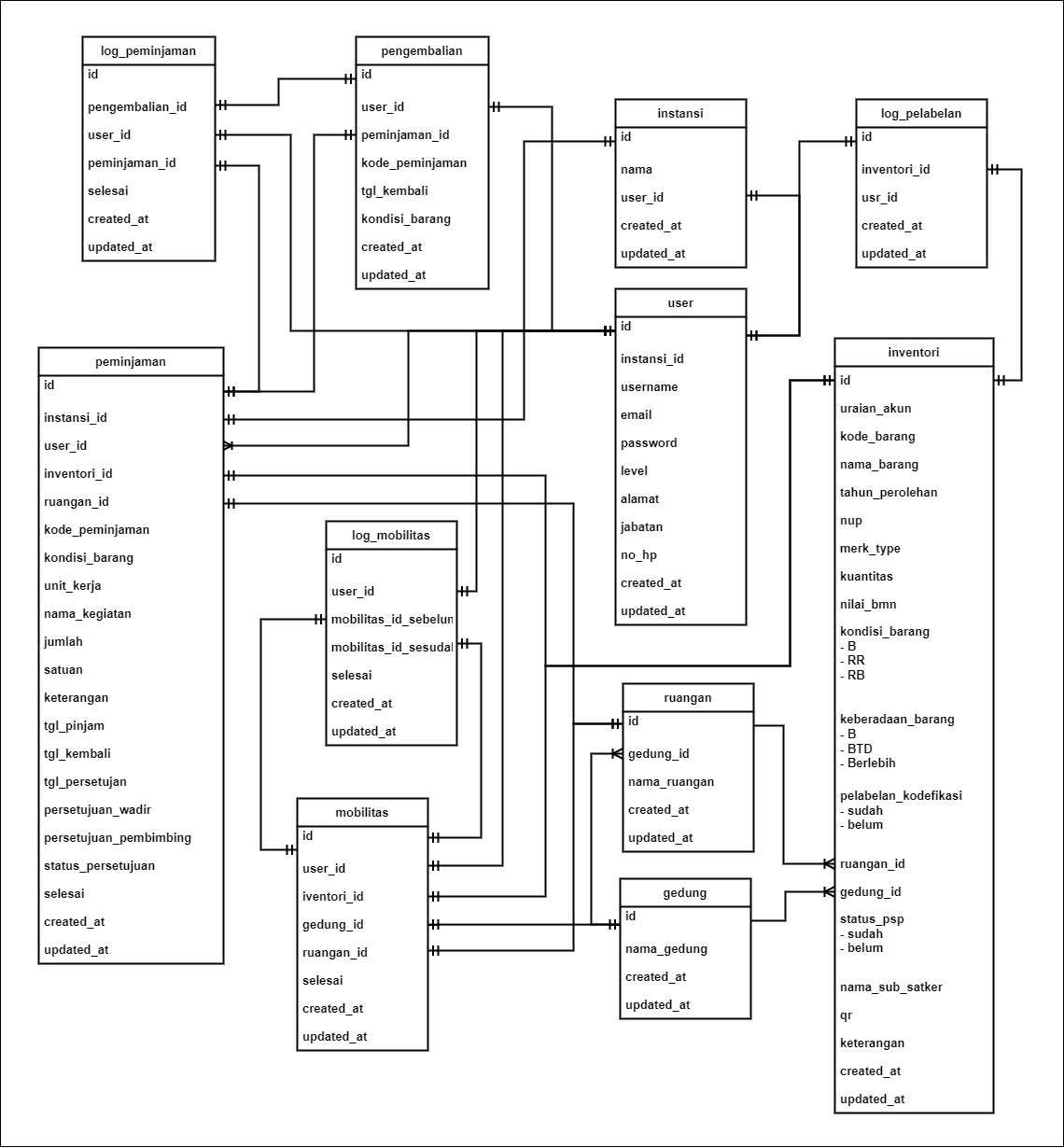
Gambar 3. 5 *Activity* diagrammobilitas



Gambar 3. 6 *Activity* diagram peminjaman

1. Class Diagram

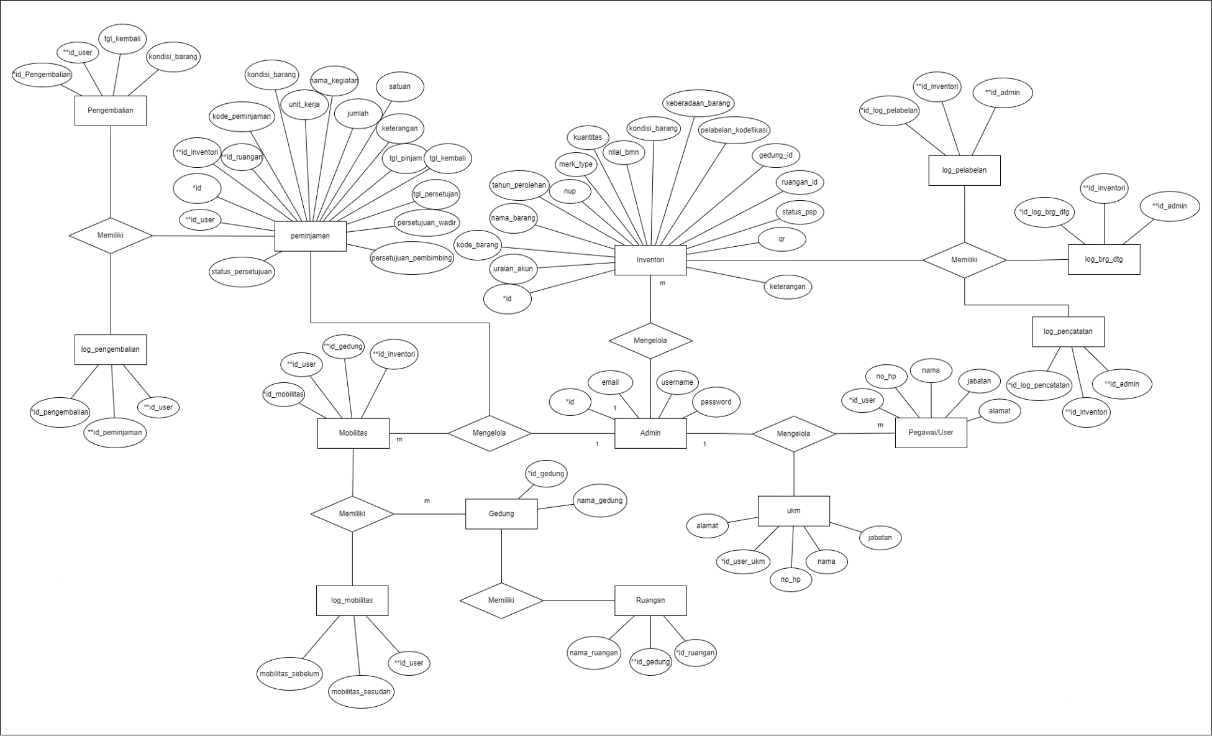
Pada gambar 3.4. di bawah ini merupakan rancangan *Class diagram* di dalam penelitian ini. Class diagram membantu para pengembang perangkat lunak dalam merencanakan struktur sistem, mengidentifikasi hubungan antar komponen, dan memahami hierarki kelas. Pada gambar 3.4 menjelaskan bahwa terdapat Class diagram yang terdiri dari 13 class dari sistem inventory barang.



Gambar 3. 7 *Class Diagram*

1. Entity Relationship Diagram (ERD)

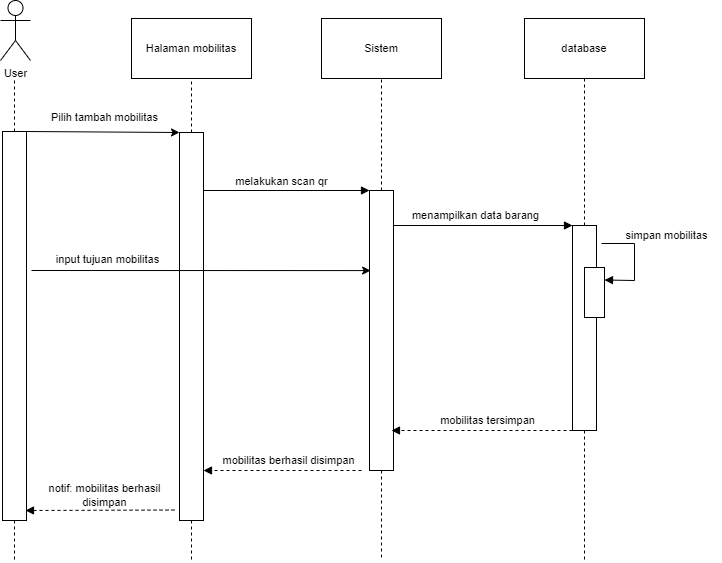
Pada perancangan aplikasi terdapat *Entity Relationship Diagram (ERD)* dari penelitian ini yang dapat dilihat pada Gambar 3.5.



Gambar 3. 8 *ERD sistem*

1. Sequence Diagram

Pada perancangan aplikasi terdapat *Sequence diagram* dari penelitian ini yang dapat dilihat pada gambar dibawah ini:



Gambar 3. 9 *Sequence diagram mobilitas*



Gambar 3. 10 *Sequence* diagram peminjaman

1. Rencana Kegiatan

Rencana kegiatan yang akan penulis lakukan dan estimasi waktu yang penulis perlukan untuk menyelesaikan penelitian ini, dapat dilihat pada Tabel 3.2

Tabel 3. 2 Tabel Rencana Kegiatan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | Maret 2023 | | | | April 2023 | | | | Mei 2023 | | | | Juni 2023 | | | | Juli 2023 | | | | Agustus 2023 | | | |
| I | II | III | IV | I | II | III | IV | I | II | III | IV | I | II | III | IV | I | II | III | IV | I | II | III | IV |
| 1 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Membuat *Backlog* Produk |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Rapat Persiapan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Spirit 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Spirit 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Spirit 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Spirit 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Spirit 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Implementasi Produk |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Penyusunan Laporan/ Penulisan Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

BAB IV  
HASIL DAN PEMBAHASAN

1. Hasil Penelitian

Hasil penelitian merupakan hasil yang diharapkan oleh penulis pada proses penelitian yang sudah berjalan. Pada bagian ini, akan diuraikan tentang proses dan hasil dari penelitian yang dilakukan dengan menggunakan metode pengembangan *Agile* menjadi sebuah Aplikasi Sistem *Inventory* barang berbasis *mobile*. Peneliti merancang Aplikasi Sistem *Inventory* barang dengan menggunakan bahasa pemrograman *Dart* serta menggunakan *framework Flutter*. Pemilihan *Dart* sebagai bahasa pemrograman dilakukan karena bahasa tersebut merupakan Bahasa pemrograman yang berbasis *Mobile* serta mempunyai keunggulan yaitu keringanan dalam proses penggunaan dan juga lebih fleksibel dalam pengembangan.

1. Data Hasil Wawancara

Langkah awal dari penelitian ini adalah sesi wawancara kepada pihak mitra yang dimana berperan sebagai *Product Owner* atau pemilik aplikasi mengenai data-data apa saja yang akan dibutuhkan oleh penulis dalam pengembangan aplikasi Sistem *Inventory* Polindra, yang mana nantinya data-data tersebut berfungsi untuk membantu peneliti supaya dapat menentukan fitur apa saja yang sesuai dengan kebutuhan dari pihak *Product Owner*. Untuk bukti dokumentasi dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Sesi Wawancara dengan Petugas Bidang *Inventory*

Setelah melakukan sesi wawancara dengan Bapak Jaya yang merupakan salah satu petugas bidang *Inventory* Politeknik Negeri Indramayu. Kami berhasil memperoleh data yang kami butuhkan, beberapa diantaranya seperti: data contoh *Inventory* milik kampus, kemudian kebutuhan fungsional, dan data lainnya.

1. Fitur Aplikasi yang Dikembangkan Berdasarkan *User Story*

*User Story* digunakan untuk menjelaskan siapa saja yang akan menjadi pengguna sistem beserta tugas dan tujuannya. Hal ini sangat penting karena daftar dari *User story* tersebut akan digunakan untuk proses pengembangan sistem selanjutnya. Berikut adalah daftar *User story* yang berhasil dikumpulkan. Berdasarkan studi literatur dan wawancara yang telah penulis lakukan, penulis menarik kesimpulan mengenai fitur-fitur pengembangan yang penulis dapatkan dari product owner. Fitur-fitur tersebut di antaranya:

Tabel 4. 1 Fitur yang dikembangkan

|  |  |  |
| --- | --- | --- |
| **Sebagai** | **Saya ingin..** | **Sehingga..** |
| Pegawai / Mahasiswa | *Login* akun | Dapat melakukan peminjaman barang |
| Melihat Profil | Akun yang sudah didaftarkan dapat disesuaikan dengan data pribadi |
| Melakukan *Scan Qr* barang | Dapat mengetahui tempat barang serta untuk mengetahui barang tersebut terdaftar atau tidak di *Inventory* polindra |
| Menambahkan barang ke dalam keranjang | Dapat memilih barang secara *multiple* |
| Melakukan perpindahan barang | Dapat melihat perpindahan barang |
| Melakukan peminjaman barang | Dapat melakukan peminjaman barang |
| Melakukan pengembalian | Dapat mengembalikan barang yang dipinjam |
| Pembimbing / Penanggung Jawab | *Login* akun | Dapat melakukan peminjaman barang |
| Melihat Profil | Akun yang sudah didaftarkan dapat disesuaikan dengan data pribadi |
| Melakukan *Scan Qr* barang | Dapat mengetahui tempat barang serta untuk mengetahui barang tersebut terdaftar atau tidak di *Inventory* polindra |
| Menambahkan barang ke dalam keranjang | Dapat memilih barang secara *multiple* |
| Melakukan perpindahan barang | Dapat melihat perpindahan barang |
| Melakukan peminjaman barang | Dapat melakukan peminjaman barang |
| Melakukan persetujuan peminjaman | Dapat melakukan persetujuan terhadap peminjaman yang dipilih |
| Melakukan pengembalian | Dapat mengembalikan barang yang dipinjam |
| Wakil Direktur 2 | *Login* akun | Dapat melakukan peminjaman barang |
| Melihat Profil | Akun yang sudah didaftarkan dapat disesuaikan dengan data pribadi |
| Melakukan *Scan Qr* barang | Dapat mengetahui tempat barang serta untuk mengetahui barang tersebut terdaftar atau tidak di *Inventory* polindra |
| Melakukan persetujuan peminjaman | Dapat melakukan persetujuan terhadap peminjaman yang dipilih |

Fitur-fitur diatas penulis usulkan sebagai poin pengembangan di aplikasi SIP. Namun, tidak sebatas fitur-fitur tersebut yang akan kami kembangkan, namun juga

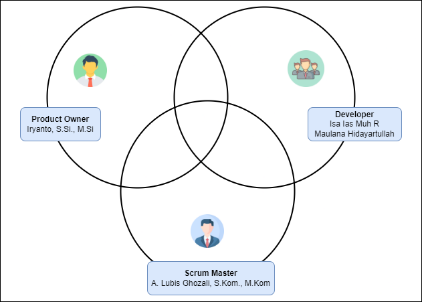
ada beberapa fitur lainnya sesuai dengan permintaan dari *product owner*. Fitur-fitur lainnya tersebut akan penulis tuangkan pada product *Backlog*. Untuk menunjang kebutuhan dari fitur-fitur tersebut diperlukan dashboard.

1. Metode *Agile Scrum*

Dalam mengembangkan aplikasi SIP diperlukan metode pengembangan perangkat lunak yang memadai untuk mengembangkan dan mengelola produk kompleks dengan lebih cepat, adaptif, dan efektif. Hal tersebut diperlukan dengan mempertimbangkan waktu skripsi yang tidak terlalu lama, dan memepertimbangkan pengembangan aplikasi lebih lanjut diluar cakupan skripsi. Maka dari itu, kami mencoba untuk menerapkan motode pengembangan *Agile* dengan framework pengembangan scrum yang terkenal dengan kehandalannya.

Pengimplementasian scrum pada pelaksanaan skripsi kami, diperlukan adaptasi atau penyesuaian. Penyesuaian tersebut diperlukan dari pertimbangan-pertimbangan sebagai berikut:

1. Waktu pelaksanaan berbarengan dengan waktu perkuliahan.
2. Pelaksanaan dalam ruang lingkup skripsi.



Gambar 4. 2 Komposisi Tim Scrum

Setelah mempertimbangkan hal-hal di atas dan berdiskusi bersama beberapa dosen, menghasilkan ide adaptasi scrum yang digambarkan pada gambar 4.2 di bawah atas.

*Product Owner* merupakan orang yang memiliki visi mengenai apa yang akan dilakukan, buat, atau capai. Pada proyek ini yang bertindak sebagai *product owner* yaitu Bapak Iryanto, S.Si., M.Si yang merupakan juga pembimbing skripsi kami. Beliau salah satu orang yang mempunyai visi terhadap awal dari perkembangan dan keberlanjutan aplikasi SIP.

*Scrum Master* merupakan orang yang melatih dan memandu rekan-rekan sesama anggota tim untuk menjalankan pendekatan *Scrum*, sekaligus membantu tim menyingkirkan apa saja yang memperlambat mereka. Untuk *scrum master* pada *project* pembuatan aplikasi SIP ini juga merupakan dosen pembimbing skripsi rekan saya penulis yaitu Bapak Bapak Iryanto, S.Si., M.Si.

Tim *Developer* merupakan orang-orang yang akan melaksanakan pekerjaan untuk mencapai visi yang telah disepakati bersama. Tim *Developer* beranggotakan orang-orang yang mempunyai keahlian-keahlian yang dibutuhkan. *Developer* pada *project* aplikasi umi ini terdiri dari 2 (dua) orang yang mempunyai spesialisasi sebagai berikut:

1. Maulana Hidayatullah (*Mobile Developer)*
2. Isa Ias Muhammad Ramadandi (*Website Developer)*

Dari kedua orang tersebut, meskipun mempunyai spesialisasi yang berbeda-beda, kami selalu saling melengkapi/membantu satu sama lain apabila mengalami kendala. Dan hal tersebutlah yang menjadi poin tambah dari pendekatan scrum.

Selain adaptasi dari komposisi tim *scrum*, kami juga mengadaptasi proses- proses yang dilaksanakan pada aktivitas inti *scrum*. Aktivitas scrum terdiri dari *Sprint Planning, Weekly Scrum, Sprint Review, dan Sprint Retrospective*. Menurut Chris Belknap dalam diskusinya di forum Scrum (2018) dan berdasarkan Buku berjudul *SCRUM* Meningkatkan Produktivitas Dua Kali Lipat dalam Waktu Setengahnya Saja yang ditulis oleh Jeff Sutherland (2017). Dari aktivitas-aktivitas *scrum* tersebut, tim *developer* wajib mengikuti seluruh aktivitas, *Product Owner* hanya diwajibkan untuk mengikuti aktivitas *Sprint review*.

Rapat pertama dalam metodologi *Scrum* dikenal sebagai *Sprint Planning.* Kegiatan ini diadakan sebelum aktivitas lainnya dimulai. Pada saat *Sprint Planning,* tim secara kolektif memeriksa *product* *Backlog* dan melakukan estimasi terhadap tugas-tugas yang dapat diselesaikan dalam satu iterasi, yang kami sebut sebagai "*Sprint*." Kami telah menetapkan periode *Sprint* selama dua minggu. Dalam konteks *Sprint* Planning, diharapkan semua anggota pengembang hadir, dan biasanya dipandu oleh seorang *Scrum Master* yang bertindak sebagai fasilitator dan pengarah. Meskipun ada saat-saat ketika kehadiran *Scrum Master* tidak memungkinkan, yang mengakibatkan kami menjalankan *Sprint Planning* tanpa kehadiran mereka. Namun, kami tetap berkomitmen untuk melaporkan hasil dari *Sprint Planning* kepada *Scrum Master.*

Selanjutnya, Weekly Scrum merupakan inti dari kerangka kerja *Scrum*. Biasanya dilakukan setiap hari pada waktu yang sama, dengan durasi tidak lebih dari 15 menit. Pada momen ini, tim pengembang dan *Scrum Master* berkumpul untuk menjawab tiga pertanyaan berikut:

* Apa yang anda lakukan kemarin untuk membantu tim menyelesaikan *Sprint*?
* Apa yang akan anda lakukan hari ini untuk membantu tim menyelesaikan *Sprint*?
* Apa saja kendala yang menghambat anda atau tim untuk mencapai target *Sprint*?

Pada aktivitas Weekly *Scrum* ini, kami menyesuaikan dengan waktu kami selaku mahasiswa, sehingga kami melaksanakan weekly *scrum* di dua waktu yang berbeda yaitu pada jam 15:00 WIB atau jam 20:00 WIB. Hal tersebut dikarenakan jadwal perkuliahan yang sering kali berubah dan kepentingan dari setiap anggota tim yang dapat menghalangi terlaksananya *weekly* *scrum* pada jam tertentu. *Weekly* *Scrum* telah kami laksanakan selama 43 kali yang dimulai dari bulan Mei sampai dengan Juli 2023.

*Sprint Review* merupakan rapat yang menjadi sarana bagi tim untuk menunjukan capaian pada satu *Sprint. Sprint review* dilaksanakan seletah *Sprint* selesai. Yang boleh didemonstrasikan oleh tim hanyalah butir yang memenuhi definisi Selesai butir *Backlog* dapat dikatakan selesai. DOD yang menjadi patokan kami yaitu Definisi Selesai atau *Definition Of Done* (DOD) merupakan patokan setiap butir *Backlog* dapat dikatakan selesai. DOD yang menjadi patokan kami yaitu sebagai berikut:

1. Fungsional aplikasi telah berjalan,

2. Aplikasi telah diuji,

3. Aplikasi tidak terdapat *bug* atau *error*, dan

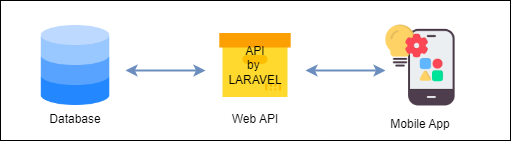
4. Aplikasi sudah dapat digunakan oleh *User*

*Sprint Retrospective* merupakan kesempatan bagi Tim *Scrum* untuk melakukan introspeksi dan menyusun rencana baru dalam rangka memperbaiki proses, guna diterapkan pada *Sprint* berikutnya.

1. Cakupan Yang Dikerjakan

Sebagai *Mobile* developer, penulis bertanggungjawab atas pengembangan aplikasi android SIP. Pada pengembangan aplikasi *Mobile*, penulis hanya mengembangkan aplikasi untuk operasi sistem android. Selain berfokus pada pembuatan kode program, penulis juga membuat desain rancangan sistem dan *User Interface.*

Aplikasi android dikembangkan dengan menggunakan bahasa pemrograman *Dart* versi 2.17.0 dan framework *Flutter* versi 3.0.0. Untuk *Database* *Management* Sistem (DBMS) kami menggunakan *MySql*. *MySql* di implemtasikan pada *backend* aplikasi android yang menggunakan *framework* *Laravel*. Dari backend aplikasi SIP dibuat sebuah *Application Programming Interface* (API) yang dapat menghasilkan response berbentuk *JavaScript Object Notation* (JSON). Dari response yang menghasilkan JSON tersebut, yang nantinya diproses lebih lanjut pada program aplikasi *Mobile*. Adapun ilustrasi proses tersebut ada pada gambar 4.3



Gambar 4. 3 Ilustrasi Integrasi Database dengan *Mobile* App

1. Artefak *Scrum*

Artefak *Scrum* adalah daftar pekerjaan yang disusun dalam rangka menyuguhkan transparansi serta memudahkan pemantauan dan juga adaptasi. *Scrum* mempunyai tiga artepak yaitu *Product Backlog, Sprint Backlog, dan Product Increment*

1. *Product Backlog*

*Product Backlog* adalah daftar berurut berisi semua butir kebutuhan yang terdapat dalam produk. *Product Backlog* bisa juga dianggap daftar persyaratan satu- satunya disebuah produk. *Product Backlog* tidak pernah komplet. *Product Backlog* awal berisi persyaratan sebagaimana sudah diketahui, dipahami, dan dibayangkan Tim *Scrum*. *Product Backlog* senantiasa berubah seiring dengan perubahan produk dan lingkungan tempat penggunaannya.

*Product Backlog* merupakan hal pertama yang dikerjakan dalam pembangunan sistem menggunakan metode *scrum*. Nantinya pada tahap ini akan dihasilkan beberapa hal seperti proses bisnis, aktor, aktivitas bisnis, struktur sistem. Isi dari product *Backlog* diperoleh dari hasil *User story* yang sudah di indentifikasi sebelumnya, dan studi literatur terkait sistem yang sedang dikembangkan.

*Product Backlog* bersifat dinamis. *Product Backlog* perlu diubah terus menerus supaya produk benar-benar pas, kompetitif, dan bernilai guna. Selama Produk masih ada, *Product Backlog* juga akan terus ada. Dalam mengembangkan aplikasi SIP, kami juga telah membuat daftar product *Backlog*. *Product Backlog* aplikasi SIP dapat dilihat pada tabel 4.2.

Tabel 4. 2 *Product Backlog*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Aktor | *Backlog* | Prioritas |
| 1 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin hak akses agar bisa masuk ke sistem | Tinggi |
| 2 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan pengecekan barang menggunakan QR code | Tinggi |
| 3 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat data pengguna | Rendah |
| 4 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin menambahkan barang ke dalam keranjang mobilitas | Sedang |
| 5 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan perpindahan lokasi barang yang ada di dalam keranjang mobilitas | Tinggi |
| 6 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat perpindahan barang yang pernah saya lakukan | Rendah |
| 7 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin menambahkan beberapa barang yang ingin saya pinjam ke dalam keranjang peminjaman | Sedang |
| 8 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan peminjaman terhadap barang yang ada di dalam keranjang peminjaman | Tinggi |
| 9 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat proses persetujuan peminjaman barang yang tadi saya tambahkan | Rendah |
| 10 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat proses peminjaman barang yang sudah di setujui | Rendah |
| 11 | Pegawai/Mahasiswa, Pembimbing/Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat peminjaman yang pernah saya lakukan | Rendah |
| 12 | Pembimbing/Penganggung jawab, | Sebagai Pembimbing / Penanggung jawab, saya ingin memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa | Sedang |
| 13 | Wakil Direktur 2 | Sebagai Wakil Direktur 2, saya ingin memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa | Sedang |

Pada product *Backlog* yang terdapat di Tabel 4.2 aktivitas nomor 1 dan 2 ditujukan untuk awalan dari sistem serta mengidentifikasi dan merancang sistem informasi berdasarkan dokumen dan data yang telah dikumpulkan. Selanjutnya aktivitas nomor 3 dan seterusnya lebih berfokus pada pengembangan sistem berbasis *Mobile* yang dalam pemrogramannya kebanyakan menggunakan bahasa *Dart*.

1. *Sprint Backlog*

*Sprint Backlog* merupakan kumpulan tugas-tugas dari *Product Backlog* yang dipilih untuk satu iterasi, juga termasuk rencana untuk melaksanakannya dan mencapai Tujuan *Sprint*. Ini adalah proyeksi dari Tim Pengembang tentang apa yang akan dianggap "selesai" pada iterasi tersebut serta pekerjaan yang diperlukan untuk mencapai tujuan tersebut. Dalam konteks aplikasi SIP, rincian tentang *Sprint* *Backlog* diuraikan lebih lanjut dalam sub bab Pelaksanaan *Sprint*.

1. *Product Increment*

*Product Increment* adalah agregasi dari item-item dalam *Product Backlog* yang telah diselesaikan selama *Sprint-Sprint* sebelumnya. Pada akhir setiap *Sprint*, inkrement baru harus memenuhi status "Selesai" yang ditetapkan oleh tim. Informasi lebih lanjut mengenai *Product Increment* dalam konteks aplikasi SIP akan dijelaskan oleh penulis dalam sub bab Pelaksanaan *Sprint*.

1. Pelaksanaan *Sprint*

Pelaksanaan setiap *Sprint* yang telah kami lakukan selama 5 *Sprint*. Dalam pelakasnaan *Sprint*, kami semua menjalankan aktivitas-aktivitas *scrum*. Di bawah ini, penulis akan sampaikan secara singkat laporan dari *Sprint* 1 sampai dengan 5 yang terdiri dari *Sprint Backlog*, dan Hasil *Sprint*. Untuk Laporan *Sprint* yang lebih lengkap dapat dilihat pada lampiran *Sprint*.

1. *Sprint* 1
   1. Sprint Goal

Dalam pelaksanaan *Sprint* 1 ini, kami menargetkan untuk beberapa modul sebagai berikut:

* + 1. Modul *login*/logout
    2. Modul cek *Inventory*
    3. Modul profil
    4. Modul dashboard
  1. Sprint Backlog

Pada *Sprint* 1 ini, kami memasukan *Backlog* yang diperkirakan dapat diselesaikan sesuai dengan skala prioritas. Butir-butir *Backlog* tersebut dilihat pada tabel 4.3

Tabel 4. 3 *Sprint* *Backlog* pada *Sprint* 1

|  |  |  |
| --- | --- | --- |
| No  *Backlog* | *Backlog* | Prioritas/  Estimasi Waktu(Jam) |
| 1. | Sebagai aktor yang dapat mengakses sistem, saya ingin hak akses agar bisa masuk ke sistem | Tinggi |
| 1.1 Membuat fitur *login* | 18 jam |
| 1.2 Testing fitur *login* |
| 1.3 Membuat fitur logout |
| 1.4 Testing Fitur logout |
| 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan pengecekan barang menggunakan QR code | Tinggi |
| 2.1 Membuat fitur cek *Inventory* | 12 jam |
| 2.2 Testing fitur cek *Inventory* |
| 3. | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat data pengguna | Tinggi |
| 3.1 Membuat fitur profil | 18 jam |
| 3.2 Testing fitur profil |
| 3.3 Membuat fitur dashboard |
| * 1. Testing fitur dashboard |

* 1. Hasil Pengerjaan *Sprint* *Backlog*

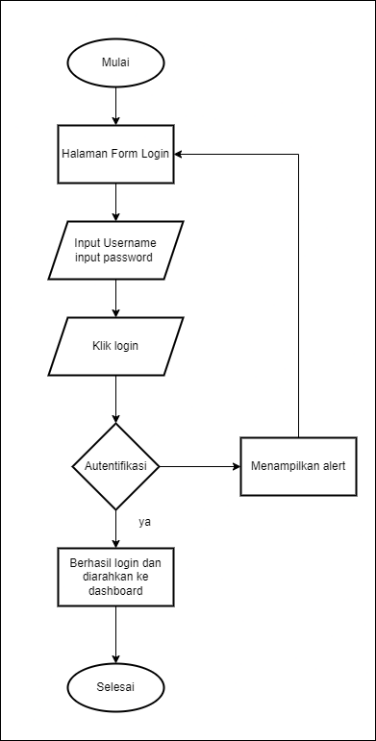
Dari semua *Backlog* yang ditargetkan spimt pertama semuanya berhasil dapat selesai pada *Sprint* 1. Sehingga pada *Sprint* 1 ini, kami baru bisa menyelesaikan 3 butir *Backlog*. Semua task *Backlog* terselesaikan, *Sprint* *Backlog* *Sprint* berikutnya melanjutkan product *Backlog* baru. Untuk butir-butir *Backlog* yang terselesaikan dan belum terselesaikan dapat dilihat pada tabel 4.4 di bawah ini. Tabel 4. 4 *Backlog* yang telah selesai dan belum selesai.

Tabel 4. 4 Hasil pengerjaan *Sprint* *Backlog*

|  |  |  |
| --- | --- | --- |
| Status | Nomor *Backlog* | Total Waktu(Jam) |
| Selesai | 1, 2 dan 3 | 56 jam |
| Belum Selesai | Semua task selesai |

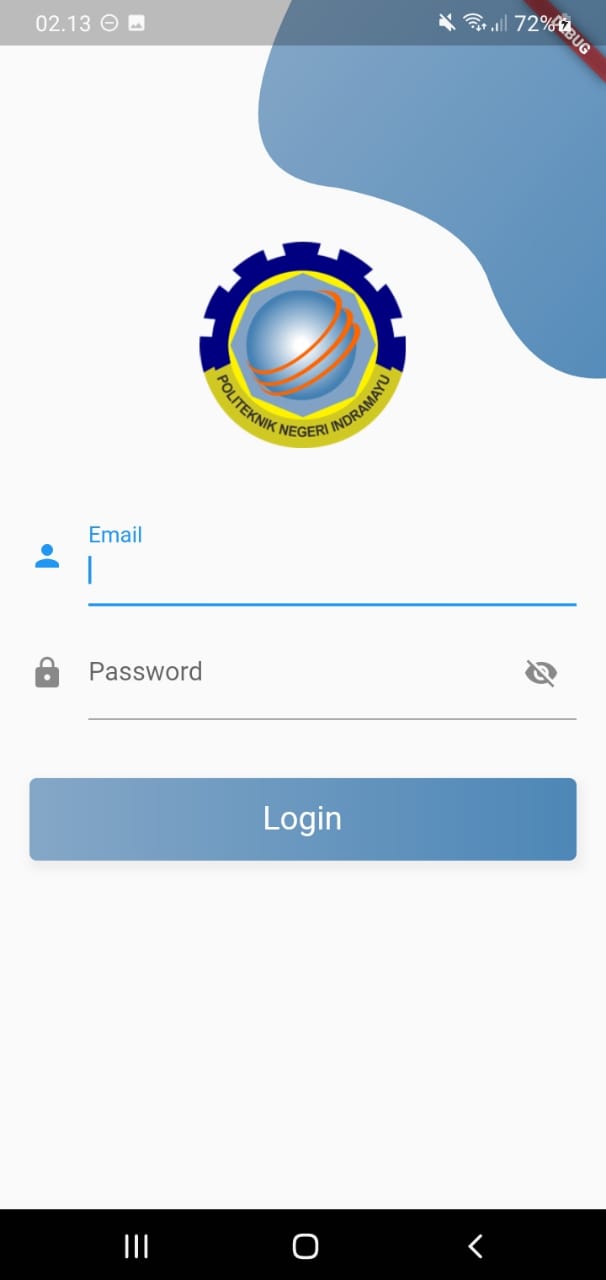
Dari butir *Backlog* yang telah terselesaikan, dihasilkanlah product increment *Sprint* 1. Berikut merupakan dokumentasi product increment yang telah dihasilkan:

1. Fitur *login*



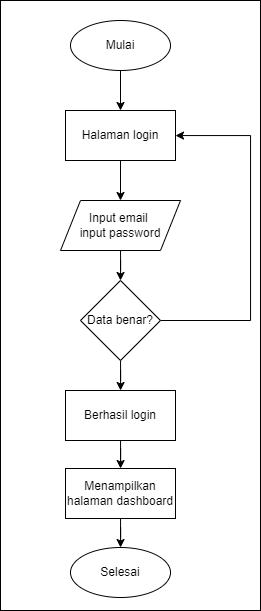
Gambar 4. 4 *Flowchart login*

Fitur *login* diperlukan *User* dapat menggunakan sistem dan menikmati fitur fitur yang ada pada aplikasi SIP *Mobile*. Tampilan halaman *login* pengguna dapat dilihat pada gambar 4.5



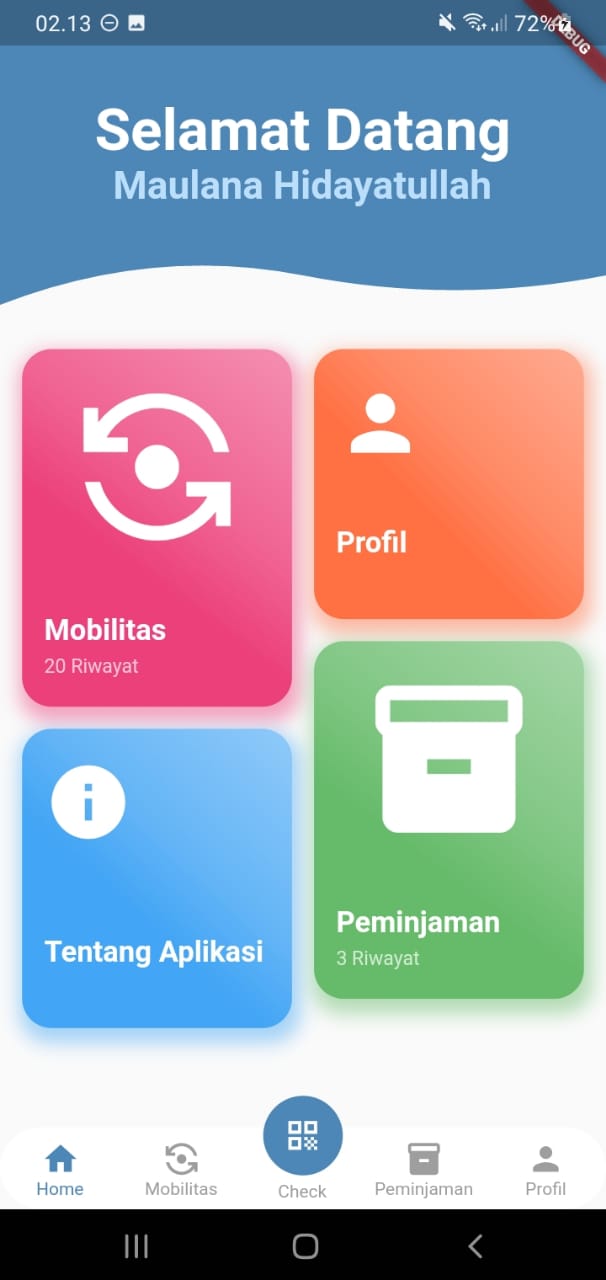
Gambar 4. 5 *Login* pengguna

1. Halaman *Dashboard*



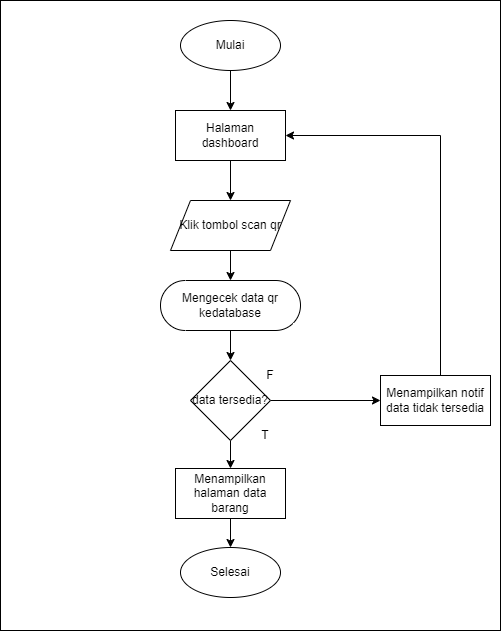
Gambar 4. 6 *Flowchart dashboard*

Halaman *dashboard* sangat diperlukan *User* dapat dengan mudah menggunakan sistem dan menikmati fitur fitur yang ada pada aplikasi SIP *Mobile*. Tampilan halaman *login* pengguna dapat dilihat pada gambar 4.5



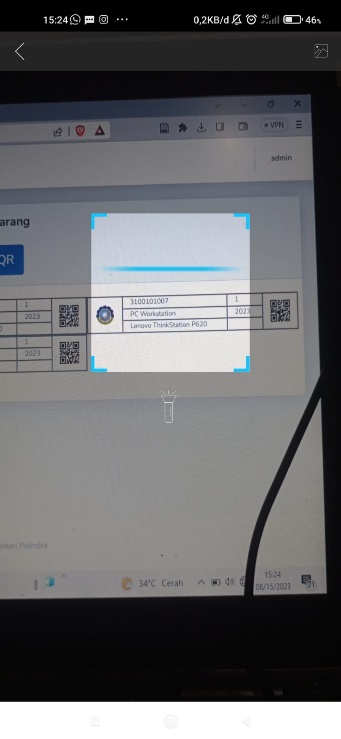
Gambar 4. 7 Halaman *Dashboard*

1. Fitur Scan Qr



Gambar 4. 8 *Flowchart Scan Qr*

Halaman Scan Qr diperlukan agar *User* dapat dengan mudah mencari data barang yang ada pada *Inventory*. Tampilan halaman *login* pengguna dapat dilihat pada gambar 4.9



Gambar 4. 9 Fitur Scan QR

* 1. Pengujian

Pengujian dilakukan dengan menggunakan metode black box testing. Metode *Blackbox Testing* adalah suatu pendekatan pengujian perangkat lunak yang fokus pada aspek fungsional dari program. Dalam metode ini, seorang tester dapat menentukan sekumpulan kondisi input dan menguji spesifikasi fungsional program tersebut. Proses pengujian *Black Box* dilakukan dengan mencoba program yang telah dibuat dan menguji setiap formnya dengan memasukkan data. Tujuan dari pengujian ini adalah untuk memastikan bahwa program berjalan sesuai dengan kebutuhan yang diperlukan. Beikut ini adalah hasil pengujian dari aplikasi SIP *Mobile* pada table 4.5

Tabel 4. 5 Tabel Pengujian *Sprint* 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Fungio**  **nalitas** | **Test case** | **Hasil yang diharapkan** | **Hasil pengujian** | **Kesim-pulan** |
| 1 | Fitur *login* | Meingputkan Email dan password dengan benar | *User* dapat *login* | *User* dapat *login* | Pengujian Berhasil |
| Meingputkan Email dan password dengan tidak sesuai | Menampilkan notif “***User*name atau Password Salah**” | Menampilkan notif “***User*name atau Password Salah**” | Pengujian Berhasil |
| 2 | Fitur Scan Qr | Menscan qr pada barcode yang sesuai | Menampilkan data barang *Inventory* yang terdaftar | Menampilkan data barang *Inventory* yang terdaftar | Pengujian Berhasil |
| Menscan qr pada barcode yang tidak sesuai | Menampilkan notif eror “**Barang tidak ditemukan**” | Menampilkan notif eror “**Barang tidak ditemukan**” | Pengujian Berhasil |
| 3 | Fitur profil pengguna | Mengklik menu profil pada halaman dashboard | Menampilkan halaman data pengguna | Menampilkan halaman data pengguna | Pengujian Berhasil |

* 1. Sprint Review

Pada *Sprint* *review* ini, kami menyampaikan hasil yang telah memenuhi definisi "Selesai" kepada *Product Owner*. Dari hasil *Sprint* *review* aplikasi telah berhasil dibuat sesuai dengan yang diharapankan. Meskipun banyak task task *Backlog* yang belum terselesaikan.

Pada *Sprint* *review* ini, kami juga meninjau kembali product *Backlog* yang ada. Hal tersebut untuk memperbaiki dan menyesuaikan product *Backlog* sesuai dengan kebutuhan. Dari hasil *Sprint* *review* ini, kami anggap telah memenuhi definisi “selesai” dari target modul awal.

* 1. Sprint Retrospective

Berikut merupakan hasil dari *Sprint* *Retrospective* yang telah kami lakukan.

* + 1. Hal baik apa yang sudah terjadi?

1. Beberapa *Backlog* berhasil diselesaikan.
2. Aktivitas apa yang belum berjalan dengan baik?
3. Testing aplikasi *Mobile* dikarenakan api web baru melakukan *deploy* pada akhir-akhir *Sprint*.
4. Apa yang perlu diperbaiki untuk *Sprint* selanjutnya?
   * + - 1. Mengurangi jumlah *task*/*Backlog* yang dikerjakan.
         2. Memperbanyak belajar untuk mengatasi bagian-bagian yang masih *error*
         3. Apabila terdapat modul yang sudah selesai, maka langsung di *deploy* agar dapat langsung menguji aplikasinya.
   1. Dokumentasi Kegiatan

|  |
| --- |
|  |

Gambar 4. 10 Dokumentasi Kegiatan

1. *Sprint* 2
   * + - 1. *Sprint* Goal

Dalam pelaksanaan *Sprint* 2 ini, kami menargetkan untuk beberapa modul sebagai berikut:

1. Modul keranjang mobilitas

2. Modul mobilitas

* + - * 1. *Sprint* *Backlog*

Pada *Sprint* 2 ini, kami memasukan *Backlog* yang diperkirakan dapat diselesaikan sesuai dengan skala prioritas. Butir-butir *Backlog* tersebut dapat dilihat pada tabel 4.6

Tabel 4. 6 *Sprint* *Backlog* pada *Sprint* 2

|  |  |  |
| --- | --- | --- |
| No  *Backlog* | *Backlog* | Prioritas/  Estimasi Waktu(Jam) |
| 4. | Sebagai aktor yang dapat mengakses sistem, saya ingin menambahkan barang ke dalam keranjang mobilitas | Tinggi |
| 4.1 Membuat fitur keranjang mobilitas | 18 jam |
| 4.2 Testing fitur keranjang mobilitas |
| 5. | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan perpindahan lokasi barang yang ada di dalam keranjang mobilitas | Tinggi |
| 5.1 Membuat fitur mobilitas | 18 jam |
| 5.2 Testing fitur mobilitas |
| 6. | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat perpindahan barang yang pernah saya lakukan | Tinggi |
| 6.1 Membuat fitur riwayat mobilitas | 18 jam |
| 6.2 Testing fitur riwayat mobilitas |

* + - * 1. Hasil Pengerjaan *Sprint* *Backlog*

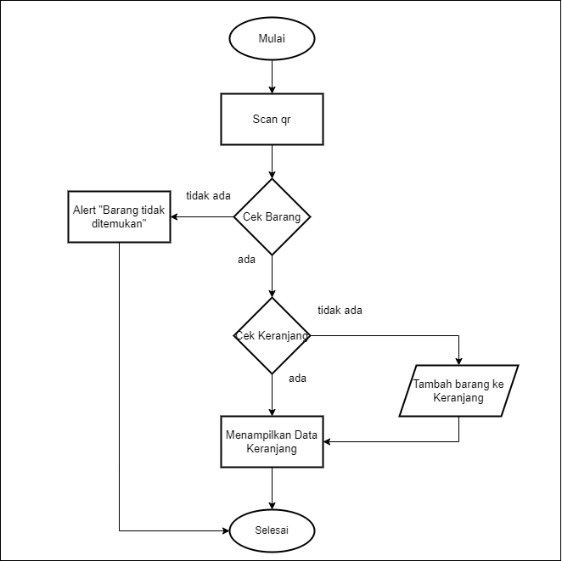
Dari semua *Backlog* yang ditargetkan spimt pertama semuanya berhasil dapat selesai pada *Sprint* 2. Sehingga pada *Sprint* 2 ini, kami baru bisa menyelesaikan 3 butir *Backlog*. Semua task *Backlog* terselesaikan, *Sprint* *Backlog* *Sprint* berikutnya melanjutkan *product* *Backlog* baru. Untuk butir-butir *Backlog* yang terselesaikan dan belum terselesaikan dapat dilihat pada tabel 4. 7 di bawah ini. Tabel 4. 7 *Backlog* yang telah selesai dan belum selesai.

Tabel 4. 7 Hasil pengerjaan *Sprint* *Backlog*

|  |  |  |
| --- | --- | --- |
| Status | Nomor *Backlog* | Total Waktu(Jam) |
| Selesai | 4,5, dan 6 | 56 jam |
| Belum Selesai | Semua task selesai |

Dari butir *Backlog* yang telah terselesaikan, dihasilkanlah product increment *Sprint* 2. Berikut merupakan dokumentasi product increment yang telah dihasilkan:

1. Fitur keranjang mobilitas



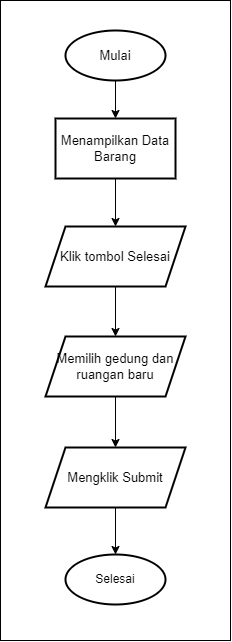
Gambar 4. 11 *Flowchart* keranjang mobilitas

Fitur keranjang mobilitas diperlukan *User* agar dapat menambahkan beberapa barang kedalam keranjang mobilitas. Tampilan fitur keranjang mobilitas dapat dilihat pada gambar 4.12

|  |
| --- |
|  |

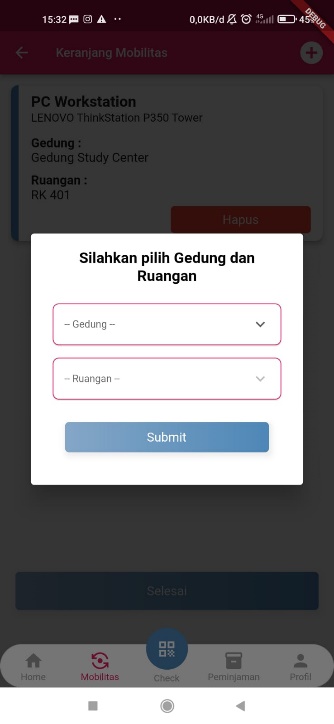
Gambar 4. 12 Fitur keranjang mobilitas

1. Fitur mobilitas



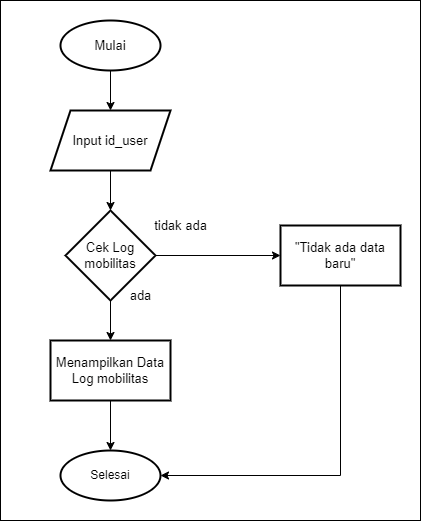
Gambar 4. 13 Flowchart keranjang mobilitas

Fitur mobilitas diperlukan *User* agar dapat melakukan perpindahan lokasi barang yang ada didalam keranjang mobilitas. Tampilan fitur mobilitas dapat dilihat pada gambar 4.9



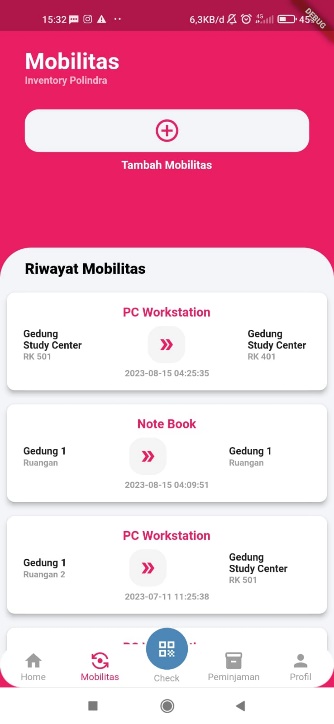
Gambar 4. 14 Fitur mobilitas

1. Fitur riwayat mobilitas



Gambar 4. 15 *Flowchart* riwayat mobilitas

Fitur riwayat mobilitas diperlukan *User* agar dapat melihat riwayat perpindahan barang yang pernah dilakukan. Tampilan fitur riwayat mobilitas dapat dilihat pada gambar 4.16



Gambar 4. 16 Fitur riwayat mobilitas

* + - * 1. Pengujian

Pengujian dilakukan dengan menggunakan metode *blackbox testing*. Metode *Blackbox Testing* adalah suatu pendekatan pengujian perangkat lunak yang fokus pada aspek fungsional dari program. Dalam metode ini, seorang tester dapat menentukan sekumpulan kondisi input dan menguji spesifikasi fungsional program tersebut. Proses pengujian *BlackBox* dilakukan dengan mencoba program yang telah dibuat dan menguji setiap formnya dengan memasukkan data. Tujuan dari pengujian ini adalah untuk memastikan bahwa program berjalan sesuai dengan kebutuhan yang diperlukan. Beikut ini adalah hasil pengujian dari aplikasi SIP *Mobile* pada table 4.8

Tabel 4. 8 Tabel Pengujian *Sprint* 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Fungio**  **-nalitas** | **Test case** | **Hasil yang diharapkan** | **Hasil pengujian** | **Kesim**  **-pulan** |
| 4 | Fitur keranjang mobilitas | Mengklik tambah keranjang mobilitas jika terdapat data barang di keranjang mobilitas | Menampilkan data barang yang terdapat pada keranjang mobilitas | Menampilkan data barang yang terdapat pada keranjang mobilitas | Pengujian Berhasil |
| Mengklik tambah keranjang mobilitas jika tidak terdapat data barang di keranjang mobilitas | Menampilkan halaman kamera untuk scan qr | Menampilkan halaman kamera untuk scan qr | Pengujian Berhasil |
| 5 | Fitur mobilitas | Memilih Gedung dan ruangan | Menampilkan notif **“Mobilitas barang berhasil”** | Menampilkan notif **“Mobilitas barang berhasil”** | Pengujian Berhasil |
| 6 | Fitur riwayat mobilitas | Mengklik riwayat mobilitas dengan kondisi tidak memiliki data | Menampilkan halaman riwayat mobilitas yang kosong | Menampilkan halaman riwayat mobilitas yang kosong | Pengujian Berhasil |

* + - * 1. *Sprint Review*

Pada *Sprint* *review* ini, kami menyampaikan hasil yang telah memenuhi definisi "Selesai" kepada *Product Owner*. Dari hasil *Sprint* *review* aplikasi telah berhasil dibuat sesuai dengan yang diharapankan.

Pada *Sprint* ini terdapat perubahan data pada tabel *Inventory*. Yakni mengikuti data yang diberikan oleh *Product Owner*.

* + - * 1. *Sprint* *Retrospective*

Berikut merupakan hasil dari *Sprint* *Retrospective* yang telah kami lakukan.

* + 1. Hal baik apa yang sudah terjadi?

1. *Backlog* *Sprint* 2 berjalan dengan baik dan lancar.
2. Beberapa *Backlog* berhasil diselesaikan.
3. Aktivitas apa yang belum berjalan dengan baik?
4. Integrasi api *Mobile* dengan web sering bermasalah. Dikarenakan hosting yang digunakan sering mati.
5. Terkendala lama waktu dikarenakan mempelajari data baru pada table *Inventory*.
6. Terdapat beberapa kesalaan kecil dari desain tampilan. Dikarenakan kurang teliti.
7. Apa yang perlu diperbaiki untuk *Sprint* selanjutnya?
   1. Mengganti layanan hosting dengan segera.
   2. Mempelajari data baru dengan segera.
      * + 1. Dokumentasi Kegiatan

|  |
| --- |
|  |

Gambar 4. 17 Dokumentasi Kegiatan

1. *Sprint* 3
   1. *Sprint* Goal

Dalam pelaksanaan *Sprint* 3 ini, kami menargetkan untuk beberapa modul sebagai berikut:

1. Modul keranjang peminjaman

2. Modul peminjaman

* 1. *Sprint* *Backlog*

Pada *Sprint* 3 ini, kami memasukan *Backlog* yang diperkirakan dapat diselesaikan sesuai dengan skala prioritas. Butir-butir *Backlog* tersebut dapat dilihat pada tabel 4.9

Tabel 4. 9 *Sprint* *Backlog* pada *Sprint* 2

|  |  |  |
| --- | --- | --- |
| No  *Backlog* | *Backlog* | Prioritas/  Estimasi Waktu(Jam) |
| 7. | Sebagai aktor yang dapat mengakses sistem, saya ingin menambahkan beberapa barang yang ingin saya pinjam ke dalam keranjang peminjaman | Tinggi |
| 7.1 Membuat fitur keranjang peminjaman | 18 jam |
| 7.2 Testing fitur keranjang peminjaman |
| 8. | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan peminjaman terhadap barang yang ada di dalam keranjang peminjaman | Tinggi |
| 8.1 Membuat fitur peminjaman | 12 jam |
| 8.2 Testing fitur peminjaman |

* 1. Hasil Pengerjaan *Sprint* *Backlog*

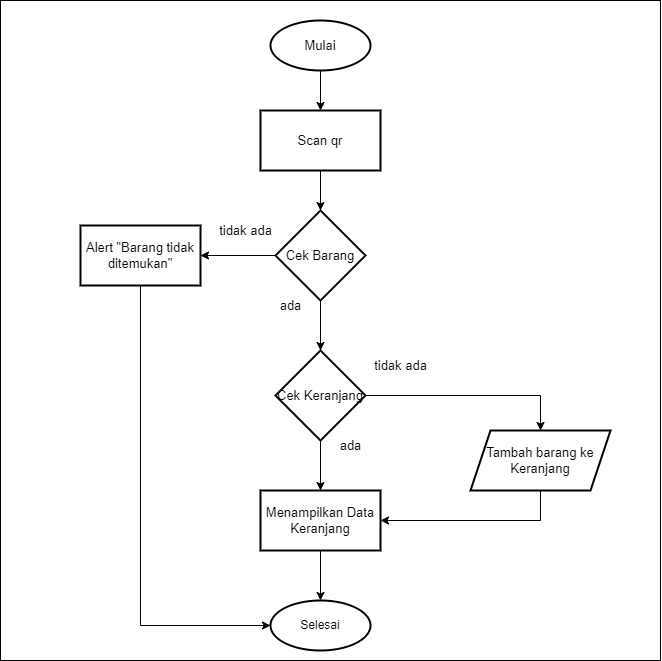
Dari semua *Backlog* yang ditargetkan spimt pertama semuanya berhasil dapat selesai pada *Sprint* 3. Sehingga pada *Sprint* 3 ini, kami menyelesaikan 2 butir *Backlog*. Semua task *Backlog* terselesaikan, *Sprint* *Backlog* *Sprint* berikutnya melanjutkan product *Backlog* baru. Untuk butir-butir *Backlog* yang terselesaikan dan belum terselesaikan dapat dilihat pada tabel 4. 10 di bawah ini. Tabel 4. 10 *Backlog* yang telah selesai dan belum selesai.

Tabel 4. 10 Hasil pengerjaan *Sprint* *Backlog*

|  |  |  |
| --- | --- | --- |
| Status | Nomor *Backlog* | Total Waktu(Jam) |
| Selesai | 7 dan 8 | 56 jam |
| Belum Selesai | Semua task selesai |

Dari butir *Backlog* yang telah terselesaikan, dihasilkanlah product increment *Sprint* 3. Berikut merupakan dokumentasi product increment yang telah dihasilkan:

1. Keranjang peminjaman



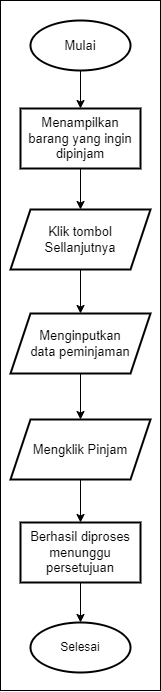
Gambar 4. 18 *Flowchart* keranjang peminjaman

Fitur keranjang peminjaman diperlukan *User* agar dapat menambahkan beberapa barang yang ingin saya pinjam ke dalam keranjang peminjaman. Tampilan fitur keranjang peminjaman dapat dilihat pada gambar 4.19



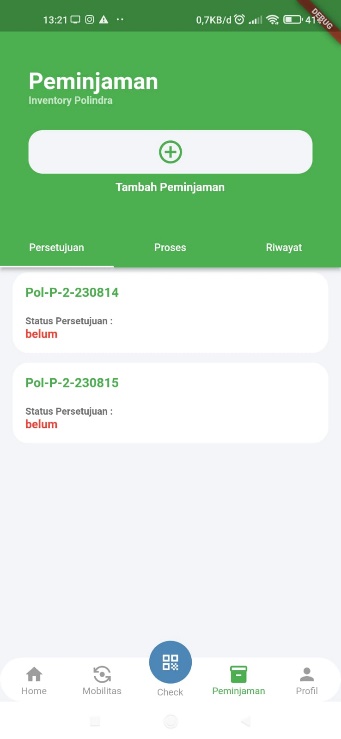
Gambar 4. 19 Fitur Keranjang Peminjaman

1. Fitur Peminjaman



Gambar 4. 20 *Flowchart* Fitur peminjaman

Fitur peminjaman diperlukan *User* agar dapat menambahkan beberapa barang kedalam keranjang mobilitas. Tampilan fitur peminjaman dapat dilihat pada gambar 4.21



Gambar 4. 21 Fitur peminjaman

* 1. Pengujian

Pengujian dilakukan dengan menggunakan metode *blackbox testing*. Metode *Blackbox Testing* adalah suatu pendekatan pengujian perangkat lunak yang fokus pada aspek fungsional dari program. Dalam metode ini, seorang tester dapat menentukan sekumpulan kondisi input dan menguji spesifikasi fungsional program tersebut. Proses pengujian *Blackbox* dilakukan dengan mencoba program yang telah dibuat dan menguji setiap formnya dengan memasukkan data. Tujuan dari pengujian ini adalah untuk memastikan bahwa program berjalan sesuai dengan kebutuhan yang diperlukan. Beikut ini adalah hasil pengujian dari aplikasi SIP *Mobile* pada table 4.11

Tabel 4. 11 Tabel Pengujian *Login*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Fungio**  **nalitas** | **Test case** | **Hasil yang diharapkan** | **Hasil pengujian** | **Kesim-pulan** |
| 7 | Fitur Keranjang Peminjama-n | Menscan qr pada barcode yang sesuai | Menampilkan data barang | Menampilkan data barang | Pengujian Berhasil |
| Menscan qr pada barcode yang tidak sesuai | Menampilkan notif “Barang tidak ditemukan” | Menampilkan notif “Barang tidak ditemukan” | Pengujian Berhasil |
| 8 | Fitur peminjam-an | Menambahkan peminjaman dengan mengisi data yang benar | Menampilkan notif “Peminjaman berhasil ditambahakan” | Menampilkan notif “Peminjaman berhasil ditambahakan” | Pengujian Berhasil |
| Menambahkan peminjaman dengan mengosongkan data | Menampilkan notif “Data tidak boleh kosong” | Menampilkan notif “Data tidak boleh kosong” | Pengujian Berhasil |

* 1. *Sprint Review*

Pada *Sprint* *review* ini, Menambah beberapa tabel pada database pengembalian, peminjaman, dan log\_peminjaman

* 1. *Sprint* *Retrospective*

Berikut merupakan hasil dari *Sprint* *Retrospective* yang telah kami lakukan.

* + 1. Hal baik apa yang sudah terjadi?
       - 1. Weekly scrum berjalan dengan baik dan lancar.
         2. Hosting sudah terbaru.
         3. Beberapa *Backlog* berhasil diselesaikan.
    2. Aktivitas apa yang belum berjalan dengan baik?
       - 1. Sama seperti *Sprint* 2. Terkendala waktu karena harus mempelajari data baru.
    3. Aplikasi baru melakukan deploy pada akhir-akhir *Sprint*.

Apa yang perlu diperbaiki untuk *Sprint* selanjutnya?

* + - * 1. Mempelajari data baru dengan segera
  1. Dokumentasi Kegiatan

|  |
| --- |
|  |

Gambar 4. 22 Dokumentasi Kegiatan

1. *Sprint* 4
   * + - 1. *Sprint* *Goal*

Dalam pelaksanaan *Sprint* 4 ini, kami menargetkan untuk beberapa modul sebagai berikut:

1. Modul proses persetujuan

2. Modul proses peminjaman

3. Moduk riwayat peminjaman

* + - * 1. *Sprint* *Backlog*

Pada *Sprint* 4 ini, kami memasukan *Backlog* yang diperkirakan dapat diselesaikan sesuai dengan skala prioritas. Butir-butir *Backlog* tersebut dapat dilihat pada tabel 4.12

Tabel 4. 12 *Sprint* *Backlog* pada *Sprint* 2

|  |  |  |
| --- | --- | --- |
| No  *Backlog* | *Backlog* | Prioritas/  Estimasi Waktu(Jam) |
| 9. | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat proses persetujuan peminjaman barang yang tadi saya tambahkan | Tinggi |
| 9.1 Membuat fitur proses persetujuan | 18 jam |
| 9.2 Testing fitur proses persetujuan |
| 10. | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat proses peminjaman barang yang sudah di setujui | Tinggi |
| 10.1 Membuat fitur proses peminjaman | 18 jam |
| 10.2 Testing fitur proses peminjaman |
| 11. | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat peminjaman yang pernah saya lakukan | Tinggi |
| 11.1 Membuat fitur riwayat peminjaman | 12 jam |
| 11.2 Testing fitur riwayat peminjaman |

* + - * 1. Hasil Pengerjaan *Sprint* *Backlog*

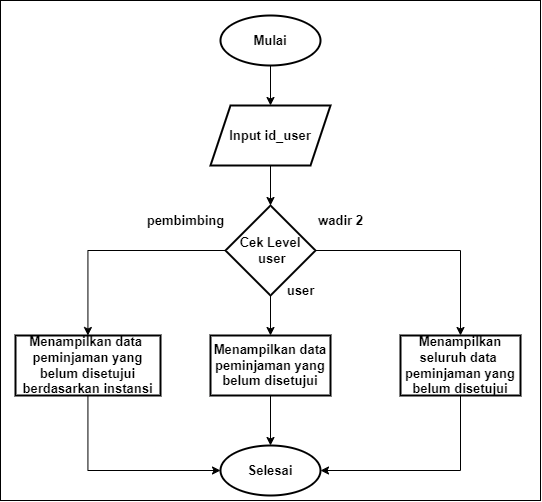
Dari semua *Backlog* yang ditargetkan pada spimt 4, 1 butir diantaranya masih belum dapat terselesaikan. Sehingga pada *Sprint* 4 ini, kami baru bisa menyelesaikan 2 butir *Backlog*. Untuk 1 butir *backlog* yang belum dapat terselesaikan, akan masuk pada *Sprint* *Backlog* di *Sprint* selanjutnya. Untuk butir-butir *Backlog* yang terselesaikan dan belum terselesaikan dapat dilihat pada tabel 4. 13 di bawah ini. Tabel 4. 13 *Backlog* yang telah selesai dan belum selesai.

Tabel 4. 13 Hasil pengerjaan *Sprint* *Backlog*

|  |  |  |
| --- | --- | --- |
| Status | Nomor *Backlog* | Total Waktu(Jam) |
| Selesai | 9 dan 10 | 56 jam |
| Belum Selesai | 11 |

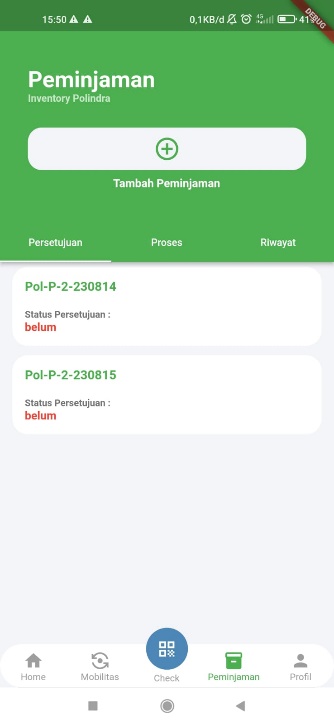
Dari butir *Backlog* yang telah terselesaikan, dihasilkanlah *product increment* *Sprint* 2. Berikut merupakan dokumentasi product increment yang telah dihasilkan:

* + 1. Fitur proses persetujuan



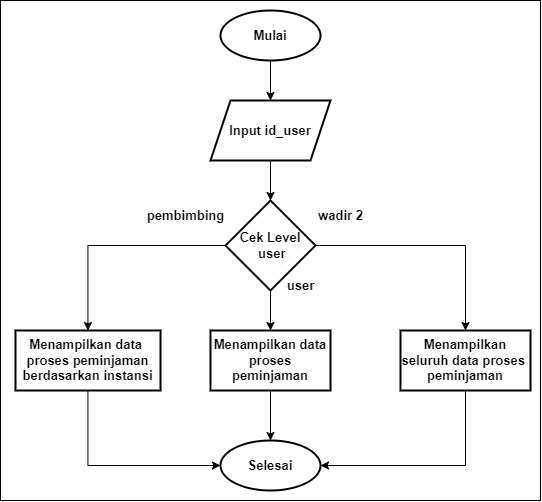
Gambar 4. 23 *Flowchart* proses persetujuan

Fitur proses persetujuan diperlukan *User* agar dapat melihat proses persetujuan peminjaman barang yang tadi saya tambahkan. Untuk memungkinkan pengguna melihat perkembangan proses persetujuan peminjaman barang yang telah saya tambahkan sebelumnya, perlu adanya langkah untuk mendapatkan persetujuan dari pengguna terlebih dahulu. Tampilan fitur proses persetujuan dapat dilihat pada gambar 4.24



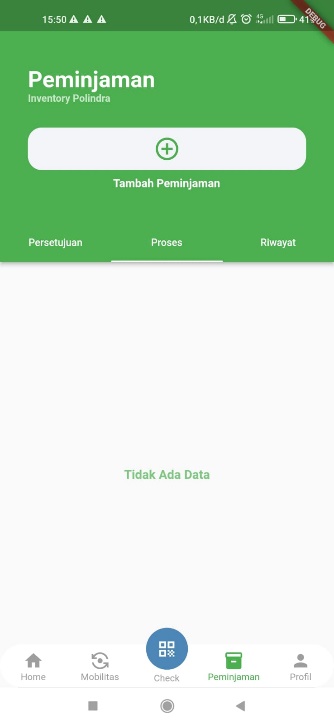
Gambar 4. 24 Fitur proses persetujuan

* + 1. Fitur proses peminjaman



Gambar 4. 25 *Flowchart* prose peminjaman

Fitur proses peminjaman diperlukan *User* agar dapat melihat proses peminjaman barang yang tadi ditambahkan. Tampilan fitur proses peminjaman dapat dilihat pada gambar 4.25



Gambar 4. 26 Proses peminjaman

* 1. Pengujian

Pengujian dilakukan dengan menggunakan metode *black box testing*. Metode *Blackbox Testing* adalah suatu pendekatan pengujian perangkat lunak yang fokus pada aspek fungsional dari program. Dalam metode ini, seorang tester dapat menentukan sekumpulan kondisi input dan menguji spesifikasi fungsional program tersebut. Proses pengujian *Black Box* dilakukan dengan mencoba program yang telah dibuat dan menguji setiap formnya dengan memasukkan data. Tujuan dari pengujian ini adalah untuk memastikan bahwa program berjalan sesuai dengan kebutuhan yang diperlukan. Beikut ini adalah hasil pengujian dari aplikasi SIP *Mobile* pada table 4.14

Tabel 4. 14 Tabel Pengujian *Sprint* 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Fungio**  **nalitas** | **Test case** | **Hasil yang diharapkan** | **Hasil pengujian** | **Kesim-pulan** |
| 9 | Fitur Persetujuan Peminjama-n | Mengklik persetujuan dengan kondisi data tersedia | Menampilkan data perstujuan peminjaman | Menampilkan data persetujuan peminjaman | Pengujian Berhasil |
| Mengklik persetujuan dengan kondisi data tidak tersedia | Tidak menampilkan data persetujuan | Tidak menampilkan data persetujuan | Pengujian Berhasil |
| 10 | Fitur Proses Peminjaman | Mengklik persetujuan dengan kondisi data tersedia | Menampilkan data proses persetujuan peminjaman | Menampilkan data proses persetujuan peminjaman | Pengujian Berhasil |
| Mengklik persetujuan dengan kondisi data tidak tersedia | Tidak Menampilkan data proses persetujuan peminjaman | Tidak menampilkan data proses persetujuan peminjaman | Pengujian Berhasil |

* 1. *Sprint Review*

Pada *Sprint* *review* ini, kami meninjau kembali *product* *Backlog* yang ada. Hal tersebut untuk memperbaiki dan menyesuaikan *product* *Backlog* sesuai dengan kebutuhan. Dikarenakan pada *Backlog* 11, tidak dapat dilakukan jika *Backlog* 12 dan 13 belum di selesaikan.

* 1. *Sprint* *Retrospective*

Berikut merupakan hasil dari *Sprint* *Retrospective* yang telah kami lakukan.

1. Hal baik apa yang sudah terjadi?
   * + 1. *Weekly* *scrum* berjalan dengan baik dan lancar.
       2. Beberapa *Backlog* berhasil diselesaikan.
3. Aktivitas apa yang belum berjalan dengan baik?
4. Masih ada *Backlog* yang belum terselesaikan.
5. Kesalahan pada urutan *Backlog*.
6. Apa yang perlu diperbaiki untuk *Sprint* selanjutnya?

Meninjau ulang *Backlog* agar tidak terjadi lagi kesalahan.

1. Dokumentasi Kegiatan

|  |
| --- |
|  |

Gambar 4. 27 Dokumentasi Kegiatan

1. *Sprint* 5
   * + - 1. *Sprint* *Goal*

Dalam pelaksanaan *Sprint* 5 ini, kami menargetkan untuk beberapa modul sebagai berikut:

* + 1. Modul riwayat peminjaman
    2. Modul persetujuan pembimbing / penanggung jawab
    3. Modul persetujuan wakil direktur 2
       - 1. *Sprint* *Backlog*

Pada *Sprint* 5 ini, kami memasukan *Backlog* yang diperkirakan dapat diselesaikan sesuai dengan skala prioritas. Butir-butir *Backlog* tersebut dapat dilihat pada tabel 4.5

Tabel 4. 15 *Sprint* *Backlog* pada *Sprint* 2

|  |  |  |
| --- | --- | --- |
| No  *Backlog* | *Backlog* | Prioritas/  Estimasi Waktu(Jam) |
| 11. | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat peminjaman yang pernah saya lakukan | Tinggi |
| 11.1 Membuat fitur riwayat peminjaman | 12 jam |
| 11.2 Testing fitur riwayat peminjaman |
| 12. | Sebagai Pembimbing / Penanggung jawab, saya ingin memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa | Tinggi |
| 12.1 Membuat fitur persetujuan pembimbing / penanggung jawab | 18 jam |
| 12.2 Testing fitur persetujuan pembimbing / penanggung jawab |
| 13. | Sebagai Wakil Direktur 2, saya ingin memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa | Tinggi |
| 13.1 Membuat fitur persetujuan wakil direktur 2 | 18 jam |
| 13.2 Testing fitur persetujuan wakil direktur 2 |

* + - * 1. Hasil Pengerjaan *Sprint* *Backlog*

Dari semua *Backlog* yang ditargetkan spimt pertama semuanya berhasil dapat selesai pada *Sprint* 5. Tabel 4. 16 *Backlog* yang telah selesai dan belum selesai.

Tabel 4. 16 Hasil pengerjaan *Sprint* *Backlog*

|  |  |  |
| --- | --- | --- |
| Status | Nomor *Backlog* | Total Waktu(Jam) |
| Selesai | 11, 12, dan 13 | 56 jam |
| Belum Selesai | Semua task selesai |

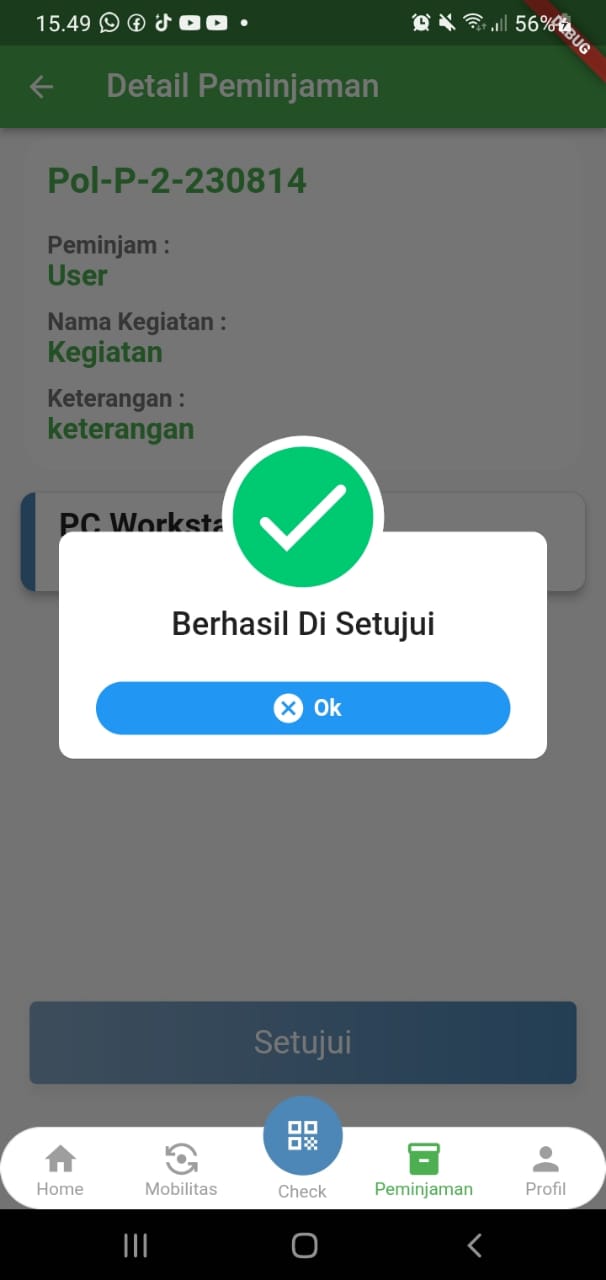
Dari butir *Backlog* yang telah terselesaikan, dihasilkanlah product increment *Sprint* 5. Berikut merupakan dokumentasi *product increment* yang telah dihasilkan:

1. Fitur persetujuan pembimbing / penanggung jawab



Gambar 4. 28 *Flowchart* persetujuan

Fitur persetujuan pembimbing / penanggung jawab diperlukan *User* agar dapat memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa. Tampilan persetujuan pembimbing / penanggung jawab dapat dilihat pada gambar 4.29



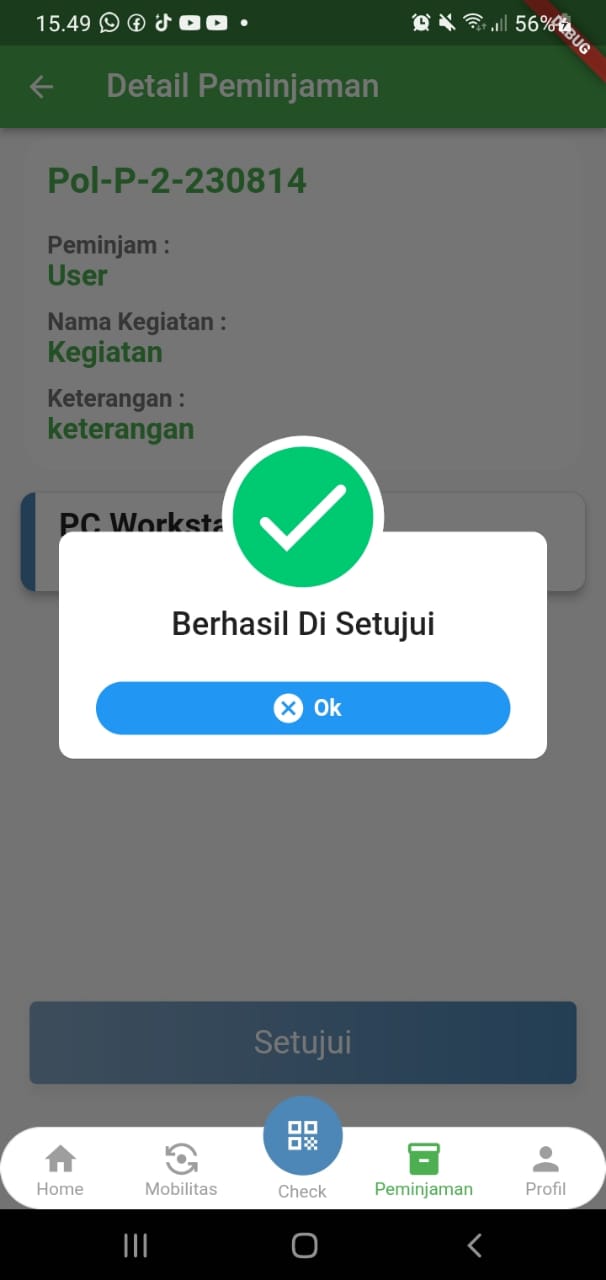
Gambar 4. 29 Fitur persetujuan pembimbing / penanggung jawab

1. Fitur persetujuan wakil direktur 2



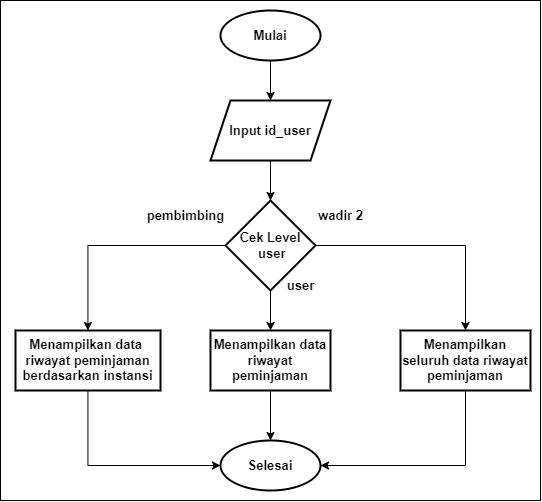
Gambar 4. 30 *Flowchart* persetujuan

Fitur persetujuan wadir 2 diperlukan *User* agar dapat memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa. Tampilan fitur keranjang mobilitas dapat dilihat pada gambar 4.31



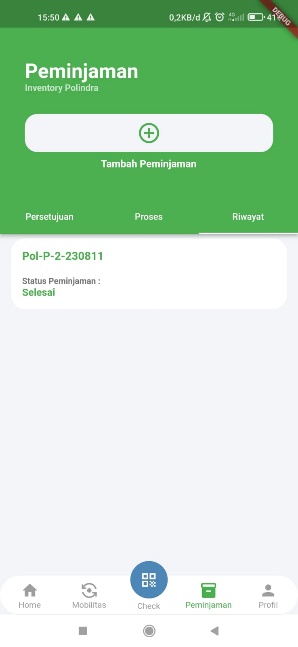
Gambar 4. 31 persetujuan wakil direktur 2

1. Fitur riwayat pengguna



Gambar 4. 32 *Flowchart* riwayat peminjaman

Fitur riwayat pengguna diperlukan *User* agar dapat melihat riwayat peminjaman yang pernah *User* lakukan. Tampilan fitur keranjang mobilitas dapat dilihat pada gambar 4.33



Gambar 4. 33 Fitur riwayat pengguna

* + - * 1. Pengujian

Pengujian dilakukan dengan menggunakan metode black box testing. Metode *Blackbox Testing* adalah suatu pendekatan pengujian perangkat lunak yang fokus pada aspek fungsional dari program. Dalam metode ini, seorang *tester* dapat menentukan sekumpulan kondisi input dan menguji spesifikasi fungsional program tersebut. Proses pengujian *Black Box* dilakukan dengan mencoba program yang telah dibuat dan menguji setiap formnya dengan memasukkan data. Tujuan dari pengujian ini adalah untuk memastikan bahwa program berjalan sesuai dengan kebutuhan yang diperlukan. Beikut ini adalah hasil pengujian dari aplikasi SIP *Mobile* pada table 4.17

Tabel 4. 17 Tabel Pengujian *Login*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | Fungsio  nalitas | Test case | Hasil yang diharapkan | Hasil pengujian | Kesim-pulan |
| 11 | Fitur Riwayat Peminjaman | Mengklik Riwayat dengan data tersedia | Menampilkan data riwayat peminjaman | Menampilkan data riwayat peminjaman | Pengujian Berhasil |
| Mengklik riwayat dengan kondisi data tidak tersedia | Tidak menampilkan data riwayat | Tidak menampilkan riwayat peminjaman | Pengujian Berhasil |
| 12 | Fitur Persetujuan Dosen/Pembimbing | Mengklik tombol persetujuan | Menampilkan notif“Berhasil Disetujui” | Menampilkan notif “Berhasil Disetujui” | Pengujian Berhasil |
| 13 | Fitur Persetujuan Wadir 2 | Mengklik tombol persetujuan | Menampilkan notif“Berhasil Disetujui” | Menampilkan notif“Berhasil Disetujui” | Pengujian Berhasil |

* + - * 1. *Sprint Review*

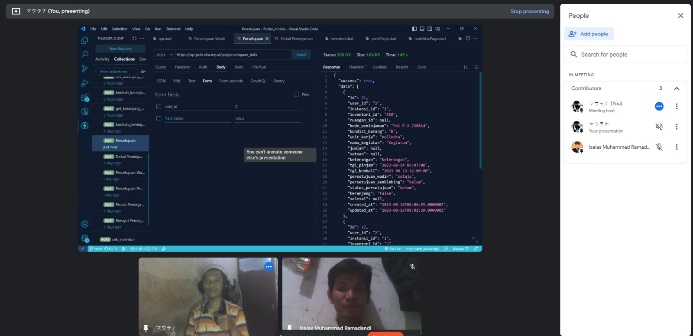
Pada *Sprint* *review* ini, kami menyampaikan hasil yang telah memenuhi definisi "Selesai" kepada *Product Owner*. Dari hasil *Sprint* *review* aplikasi telah berhasil dibuat sesuai dengan yang diharapankan.

Pada *Sprint* *review* ini, kami juga meninjau kembali *product* *Backlog* yang ada. Hal tersebut untuk memperbaiki dan menyesuaikan *product* *Backlog* sesuai dengan kebutuhan. Dari hasil *Sprint* *review* ini, kami anggap telah memenuhi definisi “selesai” dari target modul awal.

* + - * 1. *Sprint* *Retrospective*

Berikut merupakan hasil dari *Sprint* *Retrospective* yang telah kami lakukan.

1. Hal baik apa yang sudah terjadi?
2. Weekly scrum berjalan dengan baik dan lancar.
3. Semua *Backlog* berhasil diselesaikan.
5. Aktivitas apa yang belum berjalan dengan baik?
6. Pada *Sprint* ini seluruh proses telah dilaksanakan dengan baik.
7. Apa yang perlu diperbaiki untuk *Sprint* selanjutnya?
8. Terdapat beberapa fitur tambahan yang belum sempat dimasukan ked ala *Backlog* dikarenakan waktu pengerjaan skripsi yang singkat.
9. Dokumentasi Kegiatan



Gambar 4. 34 Dokumentasi Kegiatan

1. Kuesioner Pengujian

Pengujian melalui Kuesioner dilakukan kepada beberapa koresponden sebagai penguji yang menilai hasil implementasi aplikasi sebagai acuan keberhasilannya. Tujuan dari pengujian ini yaitu memastikan kesesuaian aplikasi yang telah dibuat dengan kebutuhan yang diperlukan oleh pengguna. Dimana pada skala jawaban kuesioner terdapat kriteria indeks penilaian kuesioner yang digunakan. Adapun indeks penilaian kuesioner dapat dilihat pada Tabel 4.18

Tabel 4. 18 Penilaian kuesioner

|  |  |
| --- | --- |
| **Kategori** | **Skala** |
| A (Sangat Setuju) | 4 |
| B (Setuju) | 3 |
| C (Cukup) | 2 |
| D (Kurang) | 1 |

1. Pertanyaan Kuesioner

Terdapat 5 daftar pertanyaan untuk prngguna untuk pengujian kuesioner ini meliputi keseluruhan dari komponen pengujian. Pertanyaan kuesioner tersebut dapat dilihat pada Tabel 4.9

Tabel 4. 19 Pertanyaan kuesioner

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan** | **Penilaian** | | | |
| **A** | **B** | **C** | **D** |
| 1. | Apakah Aplikasi ini mudah untuk Digunakan ? |  |  |  |  |
| 2. | Apakah aplikasi ini mudah untuk di pahami ? |  |  |  |  |
| 3. | Bagaimana menurut anda tentang keseluruhan aplikasi ini? |  |  |  |  |
| 4. | Bagaimana pendapat Anda mengenai desain tampilan aplikasi ini ? |  |  |  |  |
| 5. | Apakah setiap fitur pada aplikasi ini dapat berjalan dengan baik ? |  |  |  |  |

1. Hasil Perhitungan Kuesioner

Berikut ini adalah hasil rekapitulasi hasil kuesioner pada Aplikasi Sistem *Inventory* berbasis *Mobile.* dengan menggunakan Metode *Agile* yang telah diperoleh dapat dilihat pada Tabel 4.10

Tabel 4. 20 Hasil perhitungan kuesioner

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan** | **Penilaian** | | | |
| **A** | **B** | **C** | **D** |
| 1. | Apakah Aplikasi ini mudah untuk Digunakan ? | 13 | 6 | 1 | 0 |
| 2. | Apakah aplikasi ini mudah untuk di pahami ? | 4 | 16 | 0 | 0 |
| 3. | Bagaimana menurut anda tentang keseluruhan aplikasi ini? | 2 | 16 | 2 | 0 |
| 4. | Bagaimana pendapat Anda mengenai desain tampilan aplikasi ini ? | 1 | 17 | 2 | 0 |
| 5. | Apakah setiap fitur pada aplikasi ini dapat berjalan dengan baik ? | 2 | 14 | 4 | 0 |

Dari jawaban pengujian dengan kuesioner akan dihitung dengan menggunakan skala likert. Adapun penjelasan dari hasil pengujian dapat disimpulkan sebagai berikut:

|  |  |
| --- | --- |
| Total responden  **Skor nilai:** | = **40** |
| A (Sangat Setuju) | = 4 |
| B (Setuju) | = 3 |
| C (Cukup) | = 2 |
| D (Kurang) | = 1 |
| **Skor Tertinggi** | **= 4 x 40 = 160** |
| **Skor Terendah** | **= 1 x 40 = 40** |
| **Skor Kriteria** | **= Jumlah Responden x Skor nilai** |
|  |  |

Maka dari rumus presentase persetujuan di atas, dapat dipe*role*h hasil perhitungan kuesioner sebagai berikut:

1. Pertanyaan Pertama

Jumlah = 88,75%

1. Pertanyaan Kedua

Jumlah = 86,88%

1. Pertanyaan Ketiga

Jumlah = 80,62%

1. Pertanyaan Keempat

Jumlah = 76,25 %

1. Pertanyan kelima

Jumlah = 75%

1. Rekapitulasi Hasil Perhitungan Kuesioner

Rekapitulasi hasil perhitungan pengujian dengan kuesioner untuk semua persyaratan dapat dipe*role*h rata-rata presentase dengan menggunakan rumus seperti pada persamaan berikut:

Keterangan = Presentase Persetujuan Kategori

**=** Jumlah Presentase Nilai

**=** Jumlah Pertanyaan

Maka diperoleh semua rumus rekapitulasi hasil perhitungan keusioner sebagai berikut:

1. Kesimpulan Pengujian

Berdasarkan hasil pengujian melalui kuesioner pada Aplikasi Sistem *Inventory* berbasis *Mobile.* dengan menggunakan Metode *Agile* ini sudah berjalan dengan baik. Perolehan hasil presentase pengujian dengan menggunakan kuesioner yang dilakukan oleh 40 responden memperoleh hasil dengan nilai rata rata 81,5%

BAB V  
PENUTUP

1. Kesimpulan

Dari hasil penelitian dan pembahasan yang telah dilakukan, peneliti dapat menyimpulkan hal-hal berikut mengenai Rancang Bangun Aplikasi Sistem *Inventory* Barang Berbasis *Mobile* pada Politeknik Negeri Indramayu dengan menggunakan Metode *Agile*:

1. Berdasarkan penelitian ini, dapat disimpulkan bahwa efektivitas dan efisiensi pengelolaan Inventory barang di Politeknik Negeri Indramayu menghadapi tantangan di era modern. Kendala dan hambatan seperti kurangnya sistem terkomputerisasi, kurangnya data real-time, dan tumpang tindihnya proses pengelolaan *inventory* menjadi faktor yang mempengaruhi kinerja pengelolaan *inventory*.
2. Dalam konteks pengembangan sistem *Inventory* terkomputerisasi di Politeknik Negeri Indramayu, metode Agile menawarkan pendekatan yang sesuai untuk menghadapi perubahan berkepanjangan. Fleksibilitas metode ini memungkinkan tim pengembangan untuk merespon perubahan kebutuhan dan tuntutan dengan lebih efektif.
3. Saran

Dalam setiap perangkat lunak ini masih terdapat kelemahan dan kekurangan, oleh karena itu peneliti menyarankan untuk peneliti selanjutnya agar dapat mempertimbangkan saran yang diberikan agar optimasi parameter dapat menemukan nilai yang lebih optimal. Adapun Saran – saran tersebut antara lain sebagai berikut :

1. Untuk kedepannya diharapkan Aplikasi Sistem *Inventory* Barang berbasis *Mobile* ini berkembang dengan fitur – fitur yang lebih bermanfaat bagi pengguna.
2. Dilakukan pengembangan Aplikasi di *Platform ios* agar pengguna *ios* juga dapat menggunakan Aplikasi Sistem *Inventory* Barang pada Politeknik Negeri Indramayu.

DAFTAR PUSTAKA

Arsyad, M. N., & Lestari, D. E. G. (2020). Efektifitas penggunaan media *Mobile* learning berbasis android terhadap hasil belajar mahasiswa ikip budi utomo malang. *Agastya: Jurnal Sejarah Dan Pembelajarannya*, *10*(1), 89–105.

Azis, N., Pribadi, G., & Nurcahya, M. S. (2020). Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android. *Ikraith-Informatika*, *4*(3), 1–5.

Friandika Utamandani, V. (2021). *Penerapan Algoritma Boyer Moore Pada Pengarsipan Dokumen Di Kua Kecamatan Punung Kabupaten Pacitan Jawa Timur* (Phd Thesis). Universitas Muhammadiyah Ponorogo.

Gosal, R., & Rustam, A. (2022). Perancangan Sistem Informasi *Inventory* Berbasis Web Pada Gudang Di Pt. Spin Warriors. *Aisyah Journal Of Informatics and Electrical Engineering*, *4*(1), 27–32.

Grebić, B., & Stojanović, A. (2021). Application of the Scrum framework on projects in IT sector. *Eur. Proj. Manag. J*, *11*(2), 37–46.

Hasanuddin, H., Asgar, H., & Hartono, B. (2022). Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *Jurnal Informatika Teknologi Dan Sains*, *4*(1), 8–14.

Listiyan, E., & Subhiyakto, E. R. (2021). Rancang Bangun Sistem *Inventory* Gudang Menggunakan Metode Waterfall Studi Kasus Di Cv. Aqualux Duspha Abadi Kudus Jawa Tengah. *KONS℡ASI: Konvergensi Teknologi Dan Sistem Informasi*, *1*(1), 74–82.

Makbul, M. (2021). *Metode pengumpulan data dan instrumen penelitian*.

Mamesah, A., Rompas, S., & Katuuk, M. (2018). Hubungan Verbal Abuse Orang Tua Dengan Perkembangan Kognitif Pada Anak Usia Sekolah Di Sd Inpres Tempok Kecamatan Tompaso. *Jurnal Keperawatan*, *6*(2). https://doi.org/10.35790/jkp.v6i2.21572

Mufida, E., Rahmawati, E., & Hertiana, H. (2019). Rancang Bangun Sistem Informasi *Inventory* Pada Salonkecantikan. *Jurnal Mantik Penusa*, *3*(3).

Novendri, M. S., Saputra, A., & Firman, C. E. (2019). Aplikasi Inventaris Barang Pada Mts Nurul Islam Dumai Menggunakan Php Dan Mysql. *Lentera Dumai*, *10*(2).

Ramadhan, M. I., & Andah, B. D. (2018). Implementasi Sistem Informasi Peminjaman Aset Kampus Berbasis Web Pada Direktorat Administrasi Umum Universitas Budi Luhur Dengan Metodologi Berorientasi Obyek. *IDEALIS: InDonEsiA JournaL Information System*, *1*(4), 334–339.

Shadiq, J., Safei, A., & Loly, R. W. R. (2021). Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan *Blackbox Testing*. *Information Management For Educators And Professionals: Journal of Information Management*, *5*(2), 97–110.

Taruna, R., Ahmad, U. A., & Setiady, R. R. D. (2022). Perancangan Frontend Untuk Aplikasi Peer-to-peer Lending”salur”Berbasis Android Menggunakan *Flutter*. *EProceedings of Engineering*, *9*(3).

LAMPIRAN-LAMPIRAN

**Lampiran 1 Laporan Lengkap *Sprint***

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **AKTOR** | **DESKRIPSI** | ***SPRINT*** |
| 1 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin hak akses agar bisa masuk ke sistem | 1 |
|  |  | * 1. Membuat fitur *login*/logout |
| * + 1. Desain *Flowchart* *login*/logout |
| * + 1. Desain *UI* halaman *login* |
| * + 1. *Slicing UI* halaman *login* |
| * + 1. API *login* |
| * + 1. API logout |
| * + 1. API get *User* id berdasarkan token |
| * + 1. Integrasi API *login*/logout dengan *Mobile* |
| * + 1. Testing fitur *login*/logout |
| 2 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan pengecekan barang menggunakan QR code |
| * 1. Membuat fitur cek *Inventory* |
| * + 1. Desain *Flowchart* cek *Inventory* |
| * + 1. API cek *Inventory* |
| * + 1. Code scan qr pada *Mobile* |
| * + 1. Integrasi API cek *Inventory* dengan scan code QR *Mobile* |
| * + 1. Testing fitur cek *Inventory* |
| 3 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat data pengguna |
| * 1. Membuat fitur profil |
| * + 1. Desain *Flowchart* profil |
| * + 1. Desain UI halaman profil |
| * + 1. Slicing UI halaman profil |
| * + 1. API profil |
| * + 1. Integrasi API profil |
| * + 1. Testing fitur profil dengan *Mobile* |
| * 1. Membuat fitur dashboard |
| * + 1. Desain *Flowchart* dashboard |
| * + 1. Desain UI halaman dashboard |
| * + 1. Slicing UI halaman dashboard |
| * + 1. API dashboard |
| * + 1. Integrasi API dashboard dengan *Mobile* |
| * + 1. Testing fitur dashboard |
| 4 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin menambahkan barang ke dalam keranjang mobilitas | 2 |
| * 1. Membuat fitur keranjang mobilitas |
| * + 1. Desain *Flowchart* keranjang mobilitas |
| * + 1. Desain UI keranjang mobilitas |
| * + 1. Slicing UI keranjang mobilitas |
| * + 1. API keranjang mobilitas |
| * + 1. Integrasi API keranjang mobilitas dengan *Mobile* |
| * + 1. Testing fitur keranjang mobilitas |
| 5 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan perpindahan lokasi barang yang ada di dalam keranjang mobilitas |
| * 1. Membuat fitur mobilitas |
| * + 1. Desain *Flowchart* mobilitas |
| * + 1. Desain UI mobilitas |
| * + 1. Slicing UI mobilitas |
| * + 1. API mobilitas |
| * + 1. Integrasi API mobilitas dengan *Mobile* |
| * + 1. Testing fitur mobilitas |
| 6 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat perpindahan barang yang pernah saya lakukan |
| * 1. Membuat fitur riwayat |
| * + 1. Desain *Flowchart* riwayat mobilitas |
| * + 1. Desain UI riwayat mobilitas |
| * + 1. Slicing UI riwayat mobilitas |
| * + 1. API riwayat mobilitas |
| * + 1. Integrasi API riwayat mobilitas dengan *Mobile* |
| * + 1. Testing fitur riwayat mobilitas |
| 7 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin menambahkan beberapa barang yang ingin saya pinjam ke dalam keranjang peminjaman | 3 |
| * 1. Membuat fitur keranjang peminjaman |
| * + 1. Desain *Flowchart* keranjang peminjaman |
| * + 1. Desain UI keranjang peminjaman |
| * + 1. Slicing UI keranjang peminjaman |
| * + 1. API keranjang peminjaman |
| * + 1. Integrasi API keranjang peminjaman dengan *Mobile* |
| * + 1. Testing fitur keranjang peminjaman |
| 8 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melakukan peminjaman terhadap barang yang ada di dalam keranjang peminjaman |
| * 1. Membuat fitur peminjaman |
| * + 1. Desain *Flowchart* peminjaman |
| * + 1. Desain UI peminjaman |
| * + 1. Slicing UI peminjaman |
| * + 1. API keranjang peminjaman |
| * + 1. Integrasi API peminjaman dengan *Mobile* |
| * + 1. Testing fitur peminjaman |
| 9 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat proses persetujuan peminjaman barang yang tadi saya tambahkan | 4 |
| * 1. Membuat fitur proses persetujuan peminjaman |
| * + 1. Desain *Flowchart* proses persetujuan peminjaman |
| * + 1. Desain UI proses persetujuan peminjaman |
| * + 1. Slicing UI proses persetujuan peminjaman |
| * + 1. API proses persetujuan peminjaman |
| * + 1. Integrasi API proses persetujuan peminjaman dengan *Mobile* |
| * + 1. Testing fitur proses persetujuan peminjaman |
| 10 | Pegawai/  Mahasiswa, Pembimbing/  Penganggung jawab,  Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat proses peminjaman barang yang sudah di setujui |
| * 1. Membuat fitur proses peminjaman |
| * + 1. Desain *Flowchart* proses peminjaman |
| * + 1. Desain UI proses peminjaman |
| * + 1. Slicing UI proses peminjaman |
| * + 1. API proses peminjaman |
| * + 1. Integrasi API proses peminjaman dengan *Mobile* |
| * + 1. Testing fitur proses peminjaman |
| 11 | Pegawai / Mahasiswa, Pembimbing / Penganggung jawab, Wakil Direktur 2 | Sebagai aktor yang dapat mengakses sistem, saya ingin melihat riwayat peminjaman yang pernah dilakukan | 5 |
| * 1. Membuat fitur riwayat peminjaman |
| * + 1. Desain *Flowchart* riwayat peminjaman |
| * + 1. Desain UI riwayat peminjaman |
| * + 1. Slicing UI riwayat peminjaman |
| * + 1. API riwayat peminjaman |
| * + 1. Integrasi API riwayat peminjaman dengan *Mobile* |
| * + 1. Testing fitur riwayat peminjaman |
| 12 | Pembimbing / Penganggung jawab | Sebagai Pembimbing / Penanggung jawab, saya ingin memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa |
| * 1. Membuat fitur persetujuan pembimbing / penanggung jawab |
| * + 1. Desain *Flowchart* persetujuan pembimbing / penanggung jawab |
| * + 1. Desain UI persetujuan pembimbing / penanggung jawab |
| * + 1. Slicing UI persetujuan pembimbing / penanggung jawab |
| * + 1. API persetujuan pembimbing / penanggung jawab |
| * + 1. Integrasi API persetujuan pembimbing / penanggung jawab dengan *Mobile* |
| * + 1. Testing fitur persetujuan pembimbing / penanggung jawab |
| 13 | Wakil Direktur 2 | Sebagai Wakil Direktur 2, saya ingin memberikan persetujuan terhadap barang yang ingin di pinjam oleh Pegawai / Mahasiswa |
| * 1. Membuat fitur persetujuan wakil direktur 2 |
| * + 1. Desain *Flowchart* persetujuan wakil direktur 2 |
| * + 1. Desain UI persetujuan wakil direktur 2 |
| * + 1. Slicing UI persetujuan wakil direktur 2 |
| * + 1. API persetujuan wakil direktur 2 |
| * + 1. Integrasi API persetujuan wakil direktur 2 |
| * + 1. Testing fitur persetujuan wakil direktur 2 |

**Lampiran 2 Testing**

|  |  |
| --- | --- |
| **Identifikasi** | TC-1 |
| **Kasus uji** | Fitur *login* |
| **Tujuan** | Memeriksa apakah dapat *login* dengan data yang benar |
| **Skenario** | 1. Masukkan *User*name dan password  2. Klik tombol *login* |
| **Hasil yang diharapkan** | *User* dapat *login* |
| **Hasil pengujian** | *User* dapat *login* |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-2 |
| **Kasus uji** | Fitur *login* |
| **Tujuan** | Memeriksa apakah dapat *login* dengan data yang tidak sesuai |
| **Skenario** | 1. Masukkan *User*name dan password yang salah  2. Klik tombol *login* |
| **Hasil yang diharapkan** | Menampilkan notif “***User*name atau Password Salah**” |
| **Hasil pengujian** | Menampilkan notif “***User*name atau Password Salah**” |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-3 |
| **Kasus uji** | Fitur Scan Qr |
| **Tujuan** | Memeriksa apakah dapat mencsan qr dengan data yang benar |
| **Skenario** | 1. Menekan tombol scan qr  2. Mengarahkan kamera ke kode qr yang sesuai dari website |
| **Hasil yang diharapkan** | Menampilkan data barang *Inventory* yang terdaftar |
| **Hasil pengujian** | Menampilkan data barang *Inventory* yang terdaftar |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-4 |
| **Kasus uji** | Fitur Scan Qr |
| **Tujuan** | Memeriksa apakah dapat mencsan qr pada barcode yang tidak sesuai |
| **Skenario** | 1. Menekan tombol scan qr  2. Mengarahkan kamera ke kode qr yang tidak sesuai dari website |
| **Hasil yang diharapkan** | Menampilkan notif eror “**Barang tidak ditemukan**” |
| **Hasil pengujian** | Menampilkan notif eror “**Barang tidak ditemukan**” |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-4 |
| **Kasus uji** | Fitur profil pengguna |
| **Tujuan** | Memeriksa apakah fitur profil pengguna dapat berfungsi dengan baik |
| **Skenario** | 1. Menekan menu profil |
| **Hasil yang diharapkan** | Menampilkan halaman data pengguna |
| **Hasil pengujian** | Menampilkan halaman data pengguna |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-5 |
| **Kasus uji** | Fitur keranjang mobilitas |
| **Tujuan** | Memeriksa apakah fitur keranjang mobilitas yang sudah diisi dapat menampilkan data |
| **Skenario** | 1. Menekan tombol tambah keranjang |
| **Hasil yang diharapkan** | Menampilkan data barang yang terdapat pada keranjang mobilitas |
| **Hasil pengujian** | Menampilkan data barang yang terdapat pada keranjang mobilitas |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-6 |
| **Kasus uji** | Fitur keranjang mobilitas |
| **Tujuan** | Memeriksa apakah fitur keranjang mobilitas yang belum diisi tidak menampilkan data |
| **Skenario** | 1. Menekan tombol tambah keranjang |
| **Hasil yang diharapkan** | Menampilkan halaman kamera untuk scan qr |
| **Hasil pengujian** | Menampilkan halaman kamera untuk scan qr |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-7 |
| **Kasus uji** | Fitur mobilitas |
| **Tujuan** | Memeriksa apakah fitur mobilitas dapat berfungsi |
| **Skenario** | 1. Memilih Gedung dan ruangan  2. Menekan tombol submit |
| **Hasil yang diharapkan** | Menampilkan notif **“Mobilitas barang berhasil”** |
| **Hasil pengujian** | Menampilkan notif **“Mobilitas barang berhasil”** |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-8 |
| **Kasus uji** | Fitur riwayat mobilitas |
| **Tujuan** | Memeriksa apakah fitur riwayat mobilitas dapat berfungsi |
| **Skenario** | 1. Menekan tombol riwayat mobilitas |
| **Hasil yang diharapkan** | Menampilkan halaman riwayat mobilitas yang kosong |
| **Hasil pengujian** | Menampilkan halaman riwayat mobilitas yang kosong |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-9 |
| **Kasus uji** | Fitur Keranjang Peminjaman |
| **Tujuan** | Memeriksa apakah fitur keranjang peminjaman dapat berfungsi jika qr kode sesuai |
| **Skenario** | 1. Menekan tombol scan qr  2. Mengarahkan kamera ke qr kode yang sesuai |
| **Hasil yang diharapkan** | Menampilkan data barang |
| **Hasil pengujian** | Menampilkan data barang |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-10 |
| **Kasus uji** | Fitur Keranjang Peminjaman |
| **Tujuan** | Memeriksa apakah fitur keranjang peminjaman dapat berfungsi jika qr kode tidak sesuai |
| **Skenario** | 1. Menekan tombol scan qr  2. Mengarahkan kamera ke qr kode yang tidak sesuai |
| **Hasil yang diharapkan** | Menampilkan notif “Barang tidak ditemukan” |
| **Hasil pengujian** | Menampilkan notif “Barang tidak ditemukan” |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-11 |
| **Kasus uji** | Fitur peminjaman |
| **Tujuan** | Memeriksa apakah peminjaman dapat berfungsi jika data sesuai |
| **Skenario** | 1. Menekan tombol tambah peminjaman  2. Megisi data yang sesuai |
| **Hasil yang diharapkan** | Menampilkan notif “Peminjaman berhasil ditambahakan” |
| **Hasil pengujian** | Menampilkan notif “Peminjaman berhasil ditambahakan” |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-12 |
| **Kasus uji** | Fitur Peminjaman |
| **Tujuan** | Memeriksa apakah fitur keranjang peminjaman dapat berfungsi jika data kosong |
| **Skenario** | 1. Menekan tombol scan qr  2. Mengosongkan data |
| **Hasil yang diharapkan** | Menampilkan notif “Data tidak boleh kosong” |
| **Hasil pengujian** | Menampilkan notif “Data tidak boleh kosong” |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-13 |
| **Kasus uji** | Fitur Persetujuan Peminjaman |
| **Tujuan** | Memeriksa apakah fitur persetujuan peminjaman dapat berfungsi dengan kondisi data tersedia |
| **Skenario** | 1. Menekan tombol persetujuan peminjaman |
| **Hasil yang diharapkan** | Menampilkan data persetujuan peminjaman |
| **Hasil pengujian** | Menampilkan data persetujuan peminjaman |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-14 |
| **Kasus uji** | Fitur Persetujuan Peminjaman |
| **Tujuan** | Memeriksa apakah fitur persetujuan peminjaman dapat berfungsi dengan kondisi data tidak tersedia |
| **Skenario** | 1. Menekan tombol persetujuan peminjaman |
| **Hasil yang diharapkan** | Tidak menampilkan data persetujuan |
| **Hasil pengujian** | Tidak menampilkan data persetujuan |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-15 |
| **Kasus uji** | Fitur Proses Peminjaman |
| **Tujuan** | Memeriksa apakah fitur proses peminjaman dapat berfungsi jika kondisi data tersedia |
| **Skenario** | 1. Menekan proses peminjaman |
| **Hasil yang diharapkan** | Menampilkan data proses persetujuan peminjaman |
| **Hasil pengujian** | Menampilkan data proses persetujuan peminjaman |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-15 |
| **Kasus uji** | Fitur Proses Peminjaman |
| **Tujuan** | Memeriksa apakah fitur proses peminjaman dapat berfungsi jika kondisi data tidak tersedia |
| **Skenario** | 1. Menekan proses peminjaman |
| **Hasil yang diharapkan** | Tidak Menampilkan data proses persetujuan peminjaman |
| **Hasil pengujian** | Tidak Menampilkan data proses persetujuan peminjaman |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-16 |
| **Kasus uji** | Fitur Riwayat Peminjaman |
| **Tujuan** | Memeriksa apakah fitur riwayat peminjaman dapat berfungsi jika kondisi data tersedia |
| **Skenario** | 1. Menekan riwayat peminjaman |
| **Hasil yang diharapkan** | Menampilkan data riwayat peminjaman |
| **Hasil pengujian** | Menampilkan data riwayat peminjaman |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-17 |
| **Kasus uji** | Fitur Riwayat Peminjaman |
| **Tujuan** | Memeriksa apakah fitur riwayat peminjaman dapat berfungsi jika kondisi data tidak tersedia |
| **Skenario** | 1. Menekan riwayat peminjaman |
| **Hasil yang diharapkan** | Tidak menampilkan data riwayat |
| **Hasil pengujian** | Tidak menampilkan data riwayat |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-18 |
| **Kasus uji** | Fitur Persetujuan Dosen/Pembimbing |
| **Tujuan** | Memeriksa apakah fitur persetujuan dapat berfungsi |
| **Skenario** | 1. Menekan tombol persetujuan |
| **Hasil yang diharapkan** | Menampilkan notif “Berhasil Disetujui” |
| **Hasil pengujian** | Menampilkan notif “Berhasil Disetujui” |
| **kesimpulan** | Pengujian Berhasil |

|  |  |
| --- | --- |
| **Identifikasi** | TC-19 |
| **Kasus uji** | Fitur Persetujuan Wadir 2 |
| **Tujuan** | Memeriksa apakah fitur riwayat peminjaman dapat berfungsi |
| **Skenario** | 1. Menekan tombol persetujuan |
| **Hasil yang diharapkan** | Menampilkan notif “Berhasil Disetujui” |
| **Hasil pengujian** | Menampilkan notif “Berhasil Disetujui” |
| **kesimpulan** | Pengujian Berhasil |

**Lampiran 3 Biodata Mahasiswa**

|  |  |
| --- | --- |
| Nama | : Maulana Hidayatullah |
| NIM | : 1905017 |
| Jenis Kelamin | : Laki-laki |
| Tempat, Tanggal Lahir | : Cirebon, 17 Januari 2001 |
| Agama | : Islam |
| Jurusan / Program Studi | : D4 Rekayasa Perangkat Lunak |
| Riwayat Pendidikan | : SMP PUI Jatibarang  SMK PGRI Jatibarang  Politeknik Negeri Indramayu |
| Alamat | : Blok Karanganyar RT. 17 RW. 02 Desa Jatibarang Kecamatan Jatibarang Kabupaten Indramayu |
| Email | : aomx361@gmail.com |
| Nama Ayah | : Nata Kartawijaya S.P. |
| Nama Ibu | : Herni Pujioktiani |
| Judul Skripsi | : Rancang Bangun Sistem *Inventory* Barang Pada Politeknik Negeri Indramayu berbasis *Mobile* dengan menggunakan metode *Agile* |
| Dosen  Pembimbing | : Iryanto, S.Si., M.Si  NIP 199008012019031014 |

|  |
| --- |
| Indramayu, 28 Agustus 2023  Penulis,  Maulana Hidayatullah  NIM 1905017 |