**APLIKASI *MOBILE* MANAJEMEN AKTIVITAS PADA**

**GABUNGAN KELOMPOK TANI SRI MAKMUR**

**MENGGUNAKAN MODEL *PROTOTYPE***

**PROPOSAL TUGAS AKHIR**



**Oleh:**

**SULHIN**

**NIM 1903060**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI INDRAMAYU**

**FEBRUARI 2022**

**HALAMAN PENGESAHAN**

**APLIKASI *MOBILE* MANAJEMEN AKTIVITAS PADA**

**GABUNGAN KELOMPOK TANI SRI MAKMUR**

**MENGGUNAKAN MODEL *PROTOTYPE***

**Disusun oleh :**

**SULHIN**

**NIM 1903060**

**Proposal Tugas Akhir disetujui oleh:**

|  |  |  |  |
| --- | --- | --- | --- |
| Calon  Pembimbing | : | Fachrul P.B.M.,  S.ST., M.Kom.  NIP 199204232018031001 | ……..……………. |

Indramayu, .. Februari 2022  
 Koordinator Program Studi  
 D3 Teknik Informatika

Fachrul P.B.M, S.ST., M.Kom.  
 NIP 199204232018031001

**DAFTAR ISI**

|  |  |
| --- | --- |
| HALAMAN JUDUL…………………………………………………………... | i |
| HALAMAN PENGESAHAN…………………………………………………. | ii |
| DAFTAR ISI…………………………………………………………………... | iii |
| DAFTAR GAMBAR………………………………………………………….. | iv |
| DAFTAR TABEL……………………………………………………………... | v |
| 1. Latar Belakang Masalah…………………………………………………….. | 1 |
| 2. Rumusan Masalah…………………………………………………………... | 1 |
| 3. Batasan Masalah…………………………………………………………….. | 2 |
| 4. Tujuan……………………………………………………………………….. | 2 |
| 5. Manfaat……………………………………………………………………… | 2 |
| 6. Landasan Teori……………………………………………………………… | 3 |
| 7. Metode Pelaksanaan………………………………………………………… | 6 |
| 8. Rencana Kegiatan…………………………………………………………… | 14 |
| DAFTAR PUSTAKA…………………………………………………………. | 14 |

**DAFTAR GAMBAR**

|  |  |
| --- | --- |
| Gambar 1 Model Iteratif………………………………………………….………. | 5 |
| Gambar 2 *Activity Diagram*……………………………………………………… | 7 |
| Gambar 3 *Use Case*………………………………………………………………. | 8 |
| Gambar 4 *Flowchart* Gapoktan…………………………………………………… | 9 |
| Gambar 5 *Flowchart* Poktan…………………………………………………….. | 10 |
| Gambar 6 *Flowchart* Petani……………………………………………………… | 11 |
| Gambar 7 *Flowchart* Pembeli…………………………………………………… | 12 |

**DAFTAR TABEL**

|  |  |
| --- | --- |
| Table 1 Rencana Kegiatan …………………………………………….………. | 8 |

1. **Latar Belakang Masalah**

Gapoktan atau Gabungan Kelompok Tani merupakan kelembaga pertanian. memfasilitasi kegiatan-kegiatan pertanian dari sektor permodalan hingga pengolahan hasil tani adalah tujuan dibuatnya Gapoktan (Indrawati, 2016). Di Desa Krasak, Kecamatan Jatibarang, Kabupaten Indramayu, Provinsi Jawa Barat terdapat Gabungan Kelompok Tani (Gapoktan) Sri Makmur. Gapoktan tersebut menaungi 8 kelompok tani. Ketua Gapoktan Sri makmur menemukan beberapa tantangan Dalam proses pengelolaannya. Salah satunya adalah Adanya subsidi pupuk kimia dari pedagang perantara ke kelompok petani. Penggunaan pupuk kimia dianggap merugikan karena dapat menyebabkan harga jual hasil tani menjadi lebih rendah dan tidak sebanding dengan biaya produksinya (Swastika & Hermanto, 2011). Selain itu penggunaan pupuk kimia juga bisa mengakibatkan kerusakan pada tanah maupun hasil tani (Pujiharto, 2010). Dengan Permasalahan tersebut Ketua Gapoktan Ingin mengedukasi kepada para kelompok tani Sri Makmur tentang peralihan penggunaan pupuk kimia ke pupuk organik secara berkelanjutan.

Kelompok tani maupun ketua Gapoktan seringkali menemukan adanya keterlambatan informasi mengenai jadwal tanam, panen, dan yang perlu diketahui dari sebagian kecil kelompok tani Sri Makmur yang telah beralih dari pupuk kimia ke pupuk organik. Hal tersebut membuat ketua gapoktan belum bisa menentukan estimasi waktu yang tepat untuk panen, serta membeli dan memasarkannya berdasarkan jumlah kebutuhan pasokan dan Berpengaruh pada kesiapan Gapoktan dalam memasarkan hasil tani organik.

Gapoktan Sri Makmur berupaya untuk membuka peluang usaha untuk mengelola penjualan hasil tani organik secara mandiri dan penjualan secara terpusat. Pemasaran hasil pertanian organik dengan beberapa media dibutuhkan usaha yang lebih untuk melakukan rekapitulasi. Kendala yang dihadapi Gapoktan adalah belum tersedia sistem yang dapat membantu pemasaran secara mandiri dan terpusat.

Melihat dari permasalahan diatas, maka di usulkan Aplikasi *Mobile* Manajemen Aktivitas. Dengan diterapkannya aplikasi tersebut diharapkan kelompok tani mendapat edukasi penggunaan pupuk organik, ketua Gapoktan maupun ketua kelompok tani dimungkinkan untuk memantau informasi panen hasil tani organik, dan dilakukan pemasaran langsung secara mandiri dan terpusat.

1. **Rumusan Masalah**

Berdasarkan latar belakang yang dijabarkan, maka diperoleh suatu rumusan permasalahan yang menjadi dasar pembuatan sistem tersebut, yakni sebagai berikut :

1. Bagaimana mengedukasi kelompok tani secara berkelanjutan tentang penggunaan pupuk organik melalui aplikasi mobile.
2. Bagaimana agar ketua Gapoktan maupun ketua kelompok tani memperoleh informasi jadwal tanam hingga panen melalui aplikasi mobile.
3. Bagaimana membuat penjualan produk tani organik secara terpusat melalui aplikasi mobile.
4. **Batasan Masalah**

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

1. Cakupan aplikasi ini hanya gapoktan (gabungan kelompok petani) Sri Makmur.
2. Aplikasi berbasis *Mobile*.
3. Sistem menggunakan bahasa *framework* Flutter, dan *database* *management system* menggunakan MySql.
4. **Tujuan**

Tujuan dari pembuatan sistem yang dibuat meliputi :

1. Mengedukasi kelompok tani secara berkelanjutan tentang penggunaan pupuk organik.
2. Ketua Gapoktan maupun ketua kelompok tani memperoleh informasi jadwal tanam hingga panen.
3. Penjualan produk hasil tani organik secara terpusat melalui aplikasi mobile.
4. **Manfaat**

Manfaat yang dapat diperoleh dengan dibuatnya sistem ini adalah :

1. Mengurangi penggunaan pupuk kimia.
2. Ketua Gapoktan maupun ketua kelompok tani Mengetahui informasi jadwal tanam hingga panen.
3. Meningkatan jangkauan pemasaran produk tani.
4. **Landasan Teori**
5. ***UML (Unified Modeling Language)***

*Unified Modelling Language* atau *UML* merupakan salah satu standar bahasa yang banyak digunakan di dunia industri untuk menggambarkan kebutuhan (*requirement*), membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (PBO).

Definisi *UML* adalah sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan, dan juga pendokumentasian sistem aplikasi. Saat ini *UML* menjadi bahasa standar dalam penulisan *blue print software* (arsitektur).

*UML* sendiri memiliki macam antara lain:

1. *Structure Diagram*

Kumpulan diagram yang berfungsi untuk menjelaskan suatu struktur statis dari sistem yang dimodelkan.

1. *Behaviour Diagram*

Kumpulan diagram yang digunakan untuk menjelaskan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

1. *Interaction diagram*

Kumpulan diagram yang berfungsi untuk menjelaskan interaksi sistem dengan sistem lain maupun antar sistem pada sebuah sistem (Rosa & M, 2011).

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

*API* merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain.

Secara struktural, *API* merupakan spesifikasi dari suatu *data structure*, *objects*, *functions*, beserta parameter-parameter yang diperlukan untuk mengakses *resource* dari aplikasi tersebut. Seluruh spesifikasi tersebut membentuk suatu *interface* yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain, dan *API* dapat digunakan dengan berbagai bahasa *programming*, ataupun hanya dengan menggunakan *URL* (*Uniform Resource Locator*) yang telah disediakan oleh suatu *website*.

API memiliki 3 macam arsitektur yang sering dipakai antara lain:

1. *RPC* (*Remote Procedure Call*)

*RPC* merupakan salah satu teknologi yang digunakan untuk membantu kinerja dari *client side* dan *server side* dalam hal komunikasi, serta dapat dilakukan dengan konsep yang sederhana. *RPC* sendiri memiliki dua jenis, yaitu *XML-RPC* dan *JSON-RPC*.

1. *REST* (*Representational State Transfer*)

*REST* merupakan arsitektur *API* yang memiliki bentuk data berupa *JSON*, dimana memiliki keuntungan dari sisi performa aplikasi yang lebih ringan jika menggunakan *JSON*. Server *REST* tidak perlu mengetahui apa pun tentang status klien dan sebaliknya. Sehingga, baik server maupun klien dapat memahami pesan apa pun yang diterima, bahkan tanpa melihat pesan sebelumnya..

1. *SOAP* (*Simple Object Access Protocol*)

*SOAP* adalah protokol ringan yang ditujukan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. Soap menggunakan teknologi *XML* untuk mendefinisikan kerangka kerja tingkat lanjut yang memungkinkan pembuatan pesan yang dapat dipertukarkan pada protokol yang berbeda.(Muhammad Robith, 2020).

1. **Flutter**

Flutter adalah salah satu *Software Development Kit* (SDK) yang kompatibel digunakan untuk beragam sistem operasi. Android, iOS, Linux, MacOS, hingga Windows dapat menjalankan aplikasi yang dikembangkan melalui Flutter. Walaupun terbilang baru, Flutter adalah juga *framework* yang cukup menarik perhatian para pengembang. Berbagai kelebihan yang dimilikinya mendorong para pengembang untuk beralih menggunakan flutter. Di bawah ini akan dijelaskan lebih lanjut cara kerja serta kelebihan yang dimiliki oleh Flutter (Bimo, 2021).

1. **Petani**

Menurut UU No.19 tahun 2013tentang Perlindungan dan Pemberdayaan Petani Pasal 1 Petani adalah warga negara Indonesia perseorangan dan/atau beserta keluarganya yang melakukan Usaha Tani di bidang tanaman pangan, hortikultura, perkebunan, dan/atau peternakan.

1. **Metode Pelaksanaan**
2. **Metode Pengembangan Perangkat Lunak**

Pengembangan perangkat lunak yang digunakan yakni model *Prototype*. Dalam model Iteratif, proses iteratif dimulai dengan implementasi sederhana dari sekumpulan kecil persyaratan perangkat lunak dan secara berulang meningkatkan versi yang berkembang hingga sistem lengkap diimplementasikan dan siap untuk digunakan.

*Prototype* adalah salah satu model sub dari model I*terative. Prototype* adalah *mockup* dari beberapa aplikasi atau semua aplikasi. Idenya adalah untuk memberi pelangan lebih *intuitive hands‐on feel* tentang seperti apa tampilan aplikasi yang telah selesai. Prototipe antarmuka pengguna yang sederhana mungkin menampilkan formulir yang berisi label, kotak teks, dan tombol menunjukkan seperti apa tampilan aplikasi yang sudah jadi (Stephens, 2015).

Model siklus hidup berulang tidak mencoba untuk memulai dengan spesifikasi persyaratan yang lengkap. Sebaliknya, pengembangan dimulai dengan menentukan dan menerapkan hanya sebagian dari perangkat lunak, yang kemudian ditinjau untuk mengidentifikasi persyaratan lebih lanjut. Proses ini kemudian diulangi, menghasilkan versi baru perangkat lunak di akhir setiap iterasi model.

Proses berulang dimulai dengan implementasi sederhana dari subset dari persyaratan perangkat lunak dan secara berulang meningkatkan versi yang berkembang sampai sistem penuh diimplementasikan. Pada setiap iterasi, modifikasi desain dibuat dan kemampuan fungsional baru ditambahkan. Ide dasar di balik metode ini adalah untuk mengembangkan sistem melalui siklus berulang (berulang) dan dalam porsi yang lebih kecil pada satu waktu (inkremental).

Ilustrasi berikut adalah representasi dari model Iterative dan Incremental:



Gambar 1. Model Iteratif

Pengembangan Iteratif dan Inkremental adalah kombinasi dari desain iteratif atau metode iteratif dan model build inkremental untuk pengembangan. “Selama pengembangan perangkat lunak, lebih dari satu iterasi siklus pengembangan perangkat lunak mungkin sedang berlangsung pada waktu yang sama.” Proses ini dapat digambarkan sebagai pendekatan “akuisisi evolusioner” atau “pembangunan bertahap”.

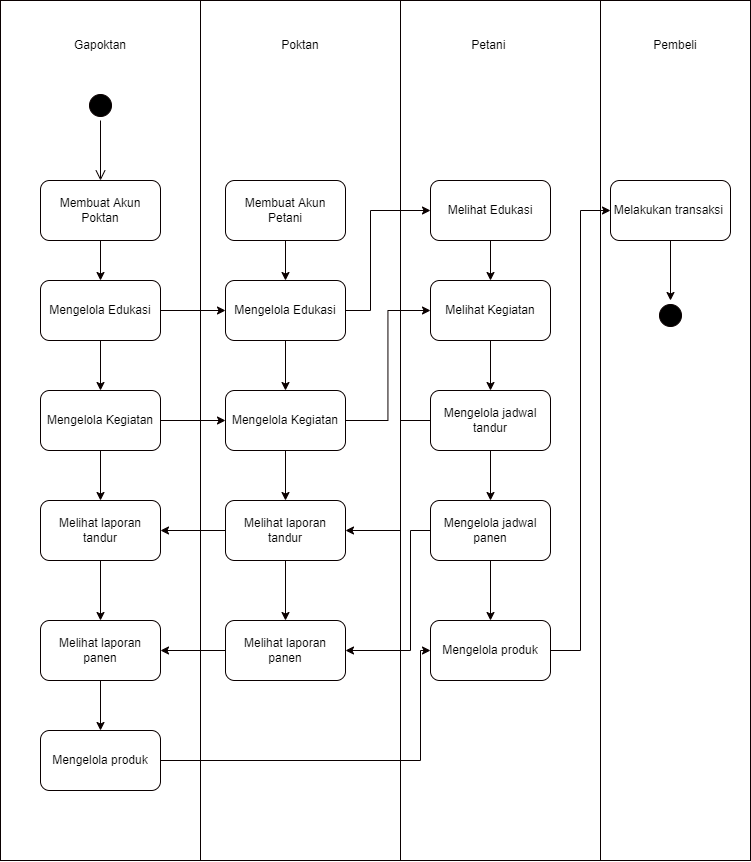
Dalam model inkremental ini, seluruh persyaratan dibagi menjadi berbagai build. Selama setiap iterasi, modul pengembangan melewati fase persyaratan, desain, implementasi, dan pengujian. Setiap rilis modul berikutnya menambahkan fungsi ke rilis sebelumnya. Proses berlanjut sampai sistem lengkap siap sesuai kebutuhan.

Kunci keberhasilan penggunaan siklus hidup pengembangan perangkat lunak berulang adalah validasi persyaratan yang ketat, dan verifikasi & pengujian setiap versi perangkat lunak terhadap persyaratan tersebut dalam setiap siklus model. Karena perangkat lunak berkembang melalui siklus yang berurutan, pengujian harus diulang dan diperpanjang untuk memverifikasi setiap versi perangkat lunak.

Seperti model SDLC lainnya, pengembangan berulang dan inkremental memiliki beberapa aplikasi khusus dalam industri perangkat lunak. Model ini paling sering digunakan dalam skenario berikut:

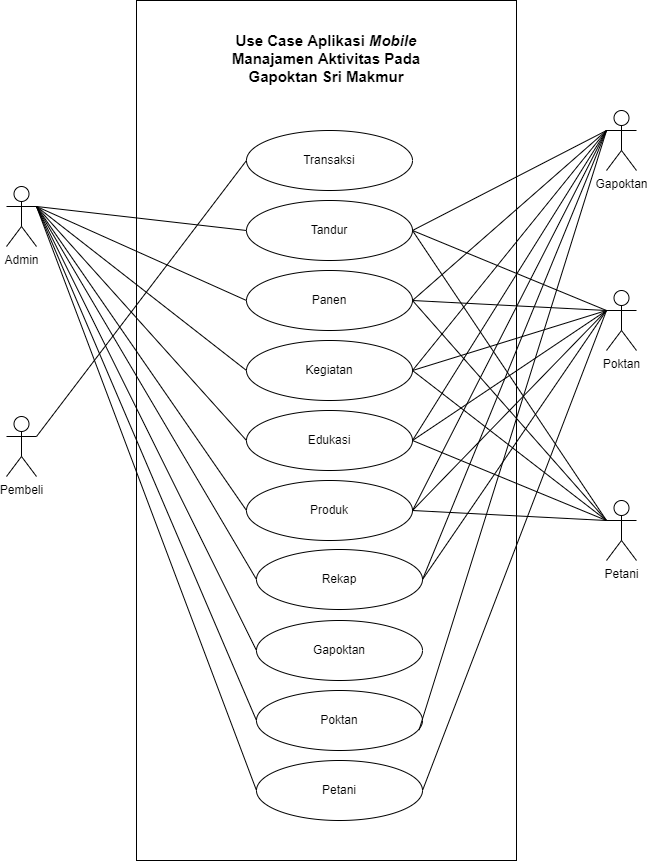
* Persyaratan sistem lengkap didefinisikan dan dipahami dengan jelas.
* Persyaratan utama harus ditentukan; namun, beberapa fungsi atau peningkatan yang diminta dapat berkembang seiring waktu.
* Ada waktu untuk kendala pasar.
* Teknologi baru sedang digunakan dan dipelajari oleh tim pengembangan saat mengerjakan proyek.
* Sumber daya dengan keahlian yang dibutuhkan tidak tersedia dan direncanakan untuk digunakan berdasarkan kontrak untuk iterasi tertentu.
* Ada beberapa fitur dan sasaran berisiko tinggi yang dapat berubah di masa depan (QNPWPAdmin, 2021).

1. **Analisis Sistem**

****

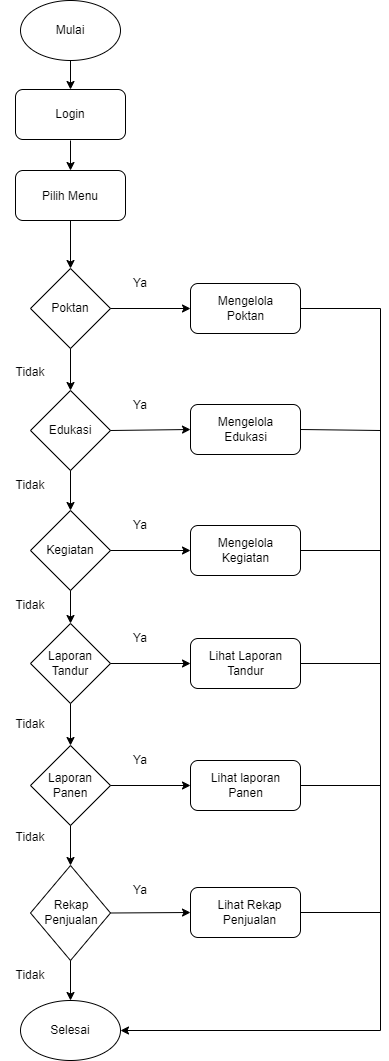
Gambar 2. *Acivity Diagram*

Gambar 2 merupakan *Activity Diagram* dari sebuah aplikasi *mobile* yang akan dibangun. Gapoktan mempunyai aktivitas membuat akun, mengelola edukasi, mengelola kegiatan, melihat laporan tandur, melihat laporan panen, dan mengelola produk. Poktan mempunyai aktivitas membuat akun petani, mengelola edukasi, mengelola kegiatan, melihat laporan tandur, dan melihat laporan panen. Petani mempunyai aktivitas melihat edukasi, melihat kegiatan, mengelola jadwal tandur, dan mengelola jadwal panen. Petani mempunyai aktivitas melihat edukasi, melihat kegiatan, mengelola jadwal tandur, mengelola jadwal panen, dan mengelola produk. Pembeli hanya mempunyai transaksi penjualan.

****

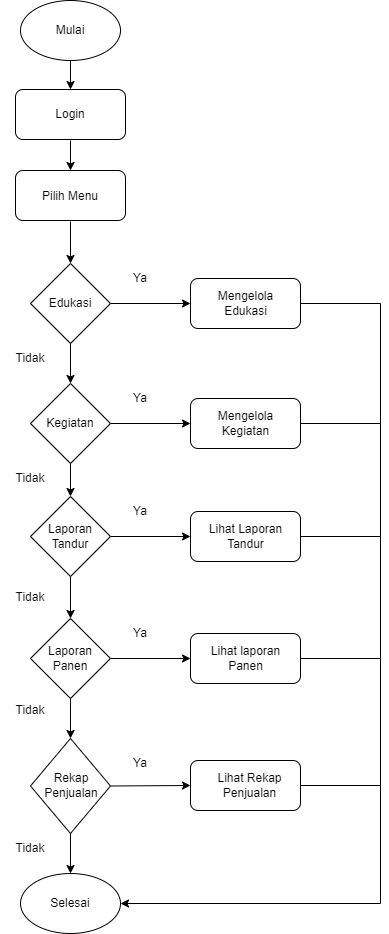
Gambar 3. *Use Case*

Gambar 3 merupakan *Use Case* dari sebuah aplikasi *website* dan *mobile* yang akan dibangun. Terdapat 5 aktor yaitu Gapoktan, Poktan, Petani, Pembeli, dan Admin. Terdapat 11 fungsional yaitu Transaksi, Tandur, Panen, Kegiatan, Edukasi, Produ, Rekap, Gapoktan, Poktan, dan Petani. Gapoktan melihat laporan tandur dan panen, mengelola kegiatan, edukasi, produk, rekap dan poktan. Poktan melihat laporan tandur dan panen, mengelola kegiatan, edukasi, produk, rekap dan petani. Petani Mengelola tandur, panen, kegiatan, edukasi, dan produk. Admin mengelola laporan tandur dan panen, mengelola kegiatan, edukasi, produk, rekap, gapoktan poktan, dan petani. Pembeli hanya dapat melakukan transaksi.



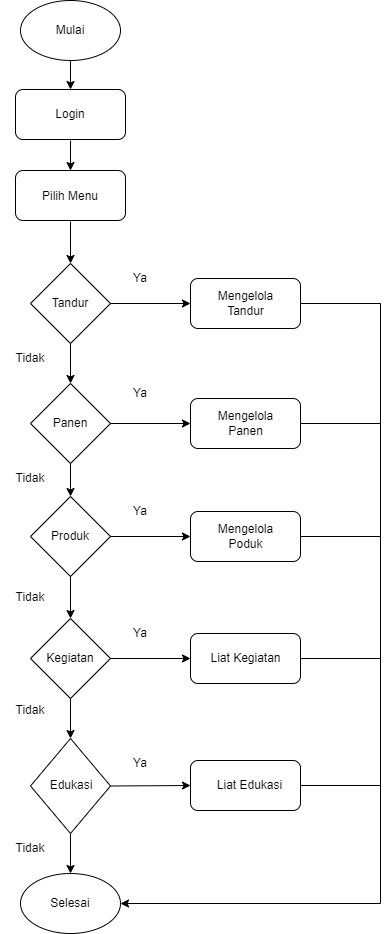
Gambar 4. Flowchart Gapoktan

Gambar 4 merupakan Flowchart Gapoktan. Pertama mulai, kemudian *login*, pilih menu, jika pilih menu poktan akan mengelola data poktan, jika pilih menu edukasi akan mengelola edukasi, jika pilih menu kegiatan akan mengelola kegiatan, jika pilih menu laporan tandur akan melihat laporan tandur, jika pilih menu laporan panen akan melihat laporan panen, jika melihat rekap penjualan akan melihat rekap penjualan, dan selesai.



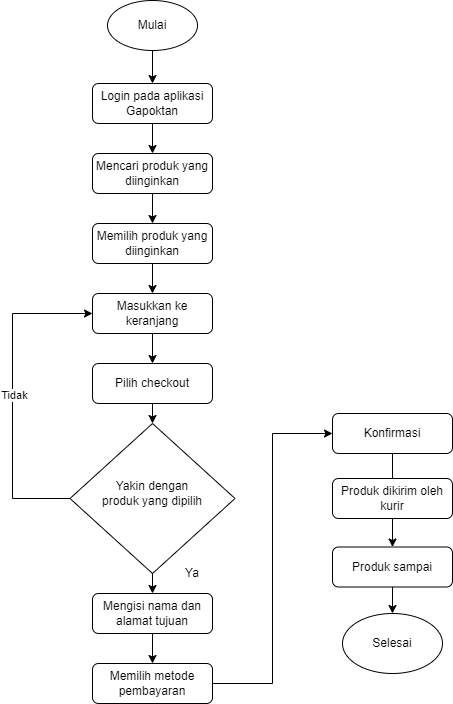
Gambar 5. Flowchart Poktan

Gambar 5 merupakan Flowchart Poktan. Pertama mulai, kemudian *login*, jika pilih menu edukasi akan mengelola edukasi, jika pilih menu kegiatan akan mengelola kegiatan, jika pilih menu laporan tandur akan melihat laporan tandur, jika pilih menu laporan panen akan melihat laporan panen, jika melihat rekap penjualan akan melihat rekap penjualan, dan selesai.



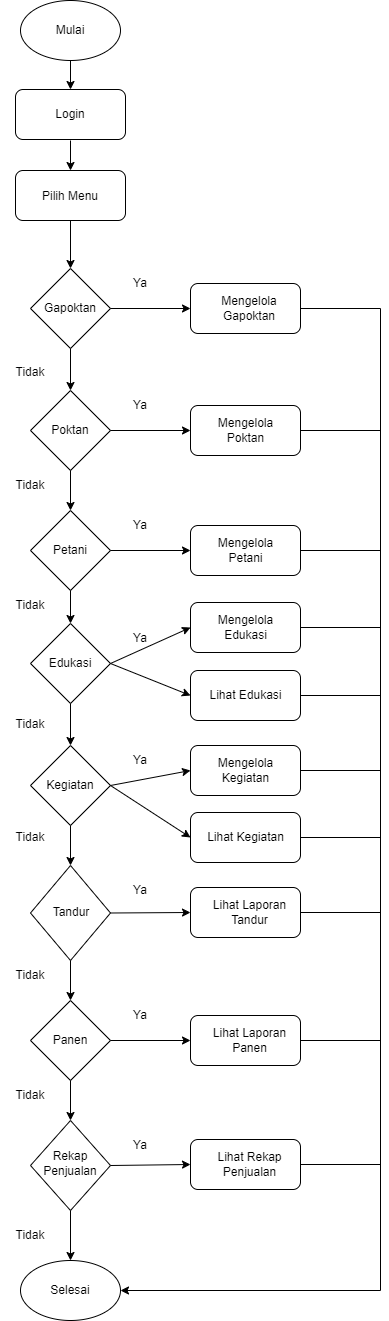
Gambar 6. Flowchart Petani

Gambar 6 merupakan Flowchart Petani. Pertama mulai, kemudian *login*, pilih menu, jika pilih menu tandur akan mengelola data tandur, jika pilih menu panen akan mengelola data panen, jika pilih menu edukasi akan melihat edukasi, jika pilih menu kegiatan akan melihat kegiatan, dan selesai.



Gambar 7. Flowchart Pembeli

Gambar 7 merupakan Flowchart Pembeli. Pertama mulai, kemudian login, mencari produk yang diinginkan, memilih produk yang diinginkan, masukkan ke keranjang, pilih *checkout* jika yakin dengan produk yang dipilih lalu mengisi nama dan alamat tujuan, memilih metode, pembayaran, konfirmasi, produk dikirim oleh kurir, produk sampai, dan selesai.

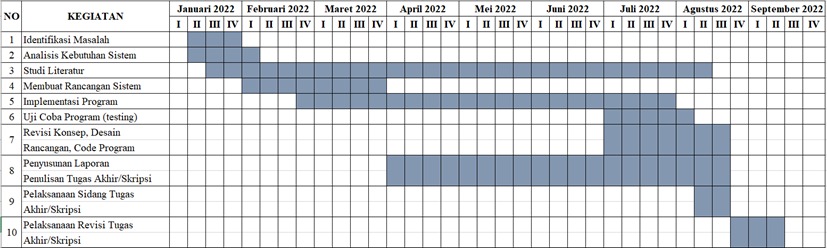


Gambar 8. Flowchart Admin

Gambar 8 merupakan Flowchart Gapoktan. Pertama mulai, kemudian *login*, pilih menu, jika pilih menu gapoktan akan mengelola data gapoktan, jika pilih menu poktan akan mengelola data poktan, jika pilih menu edukasi akan mengelola edukasi, jika pilih menu kegiatan akan mengelola kegiatan, jika pilih menu laporan tandur akan melihat laporan tandur, jika pilih menu laporan panen akan melihat laporan panen, jika melihat rekap penjualan akan melihat rekap penjualan, dan selesai.

1. **Rencana Kegiatan**

Tabel 1. Rencana Kegiatan



1. **Daftar Pustaka**

Bimo, M. (2021). *Sudah tahu cara coding dengan Flutter? Begini 4 informasi pentingnya!* Diambil kembali dari https://www.ekrut.com/media/flutter-adalah

Indrawati, N. N. (2016). Fungsi Gabungan Kelompok Tani (Gapoktan). *Departemen Antropologi, Fakultas Ilmu Sosial dan Ilmu Politik*, 335.

Muhammad Robith, A. (2020). *Mengenal penggunaan dari API beserta fungsi dan contohnya*. Diambil kembali dari https://www.sekawanmedia.co.id/blog/pengertian-api/

Pujiharto. (2010). “Kajian Pengembangan Gabungan Kelompok Tani (Gapoktan)". *Agritech, Vol. XII No*.

QNPWPAdmin. (2021). *Model Iteratif (SDLC) : Definisi, Kelebihan dan Kekurangannya*. Diambil kembali dari qnp.co.id: https://qnp.co.id/blog/model-iteratif-sdlc-definisi-kelebihan-dan-kekurangannya/

Republik Indonesia. 2013. *Undang – Undang Republik Indonesia Nomor 19 Tahun 2013 Tentang Perlindungan dan Pemberdayaan Petanian*. Kementrian Pertanian Direktorat Jendral Perkebunan. Jakarta.

Rosa, A., & M, S. (2011). Modul Pembelajaran Rekayasa . *Bandung: Modula*.

Stephens, R. (2015). Beginning Software Engineering. Indianapolis. John Wiley & Sons, Inc.

Swastika, & Hermanto. (2011). Penguatan Kelompok Tani: Langkah Awal Peningkatan Kesejahteraan Petani. *Analisis Kebijakan pertanian, Volume 9 No. 4, Desember 2011*, 371 – 390.