

Pengembangan Aplikasi Mobile Pendeteksi Penyakit Pada Tanaman Cabai Menggunakan Teknologi Clarifai

(Studi Kasus: Balai Pengkajian Teknologi Pertanian, Kecamatan Karangploso, Kota Malang)

Insan Nurzaman Bangsa Adi Pratama¹, Issa Arwani², Handoko³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹insannurzaman@student.ub.ac.id, ²issa.arwani@ub.ac.id, ³ndoro_ndoko@yahoo.com

Abstrak

Di negara-negara tropis seperti Indonesia, tanaman cabai menjadi salah satu tanaman komersial yang banyak dibudidayakan dan memiliki nilai jual tinggi sehingga menguntungkan bagi petani. Berdasarkan data dari Kementerian Pertanian, jumlah produksi cabai setiap tahunnya masih bersifat fluktuatif yaitu mengalami peningkatan atau penurunan. Adapun berdasarkan data dari Kementerian Pertanian, jumlah konsumsi dari cabai selalu mengalami kenaikan per tahunnya. Ada banyak faktor yang mempengaruhi fluktuasi tingkat produksi cabai nasional. Salah satu kendala yang sering dijumpai yaitu kurangnya pengetahuan para petani dalam mengenali jenis penyakit dan hama yang menyerang tanaman pada cabai. Dengan banyaknya pengguna perangkat *mobile* di Indonesia dapat memudahkan semua penggunanya dalam menjalankan aktivitas sehari-hari menggunakan aplikasi-aplikasi yang tersedia. Penulis memanfaatkan peluang untuk mengembangkan sebuah aplikasi *mobile* dengan sistem operasi android yang berguna untuk mendeteksi penyakit pada tanaman cabai serta pengendaliannya. Aplikasi ini memanfaatkan teknologi clarifai untuk membantu melakukan pendeteksian gambar pada input dari perangkat pengguna melalui gambar ataupun galeri. Pengembangan aplikasi ini menggunakan metode *Waterfall*. Metode ini dipilih karena kebutuhan sudah didefinisikan di awal, tidak mengalami perubahan kebutuhan, dan kebutuhan yang tidak terlalu kompleks. Pengujian yang dilakukan pada aplikasi ini adalah pengujian validasi, pengujian akurasi, pengujian, pengujian *usability* dan pengujian *compatibility*. Pada pengujian validasi semua bernilai valid. Dengan ini dapat dikatakan tingkat keberhasilan 100%. Pada pengujian akurasi didapatkan hasil nilai rata-rata kemampuan akurasi mendeteksi nama penyakit pada aplikasi ini adalah 70,59%. Hal ini dapat dikatakan bahwa sistem belum terlalu akurat. Pada pengujian *usability* didapatkan hasil 82,75%. Nilai tersebut di konversikan kedalam skala konversi kategori nilai *usability* maka akan masuk kedalam kategori nilai B atau *Excellent*. Sedangkan pada pengujian *compatibility*, aplikasi dapat berjalan sesuai dengan yang telah ditentukan yaitu Android minimal level SDK 23.

Kata kunci: *cabai, penyakit cabai, daun cabai, deteksi, identifikasi, android, clarifai, model view presenter (MVP), aplikasi perangkat bergerak, waterfall.*

Abstract

In tropical countries like Indonesia, chili is one of the most widely grown commercial crops and has a high selling value that is profitable for farmers. Based on data from the Ministry of Agriculture, the amount of chili production each year is still fluctuating, ie experiencing an increase or decrease. Based on data from the Ministry of Agriculture, the amount of consumption from chili always increases annually. There are many factors that influence the fluctuations in the level of national chili production. One obstacle that is often encountered is the lack of knowledge of farmers in recognizing the types of diseases and pests that attack plants on chilies. With so many mobile device users in Indonesia, it can be easier for all users to carry out daily activities using available applications. The author takes advantage of the opportunity to develop a mobile application with the Android operating system that is useful for detecting diseases in chili plants and its control. This application utilizes clarifai technology to help detect images on input from user devices through images or galleries. The development of this application uses the Waterfall method. This method was chosen because needs have been defined at the

beginning, do not change needs, and needs are not too complex. Tests carried out in this application are validation testing, accuracy testing, testing, usability testing and compatibility testing. In the validation test all are valid. With this it can be said that the success rate is 100%. In testing the accuracy of the results obtained an average value of the ability to detect the accuracy of the name of the disease in this application is 70.59%. This can be said that the system is not very accurate. In the usability test results obtained 82.75%. The value is converted into the conversion scale of the usability value category, it will be included in the B value or Excellent category. While in compatibility testing, the application can run in accordance with predetermined Android minimum SDK 23 level.

Keywords: *chili, chili disease, chili leaves, detection, identification, android, clarifai, model view presenter (MVP), mobile device application, waterfall.*

1. PENDAHULUAN

Di negara-negara tropis seperti Indonesia, tanaman cabai menjadi salah satu tanaman komersial yang banyak dibudidayakan dan memiliki nilai jual tinggi sehingga menguntungkan bagi petani. Tidak hanya dijual di pasaran, cabai juga biasa dijadikan sebagai bahan baku industri sehingga membuka kesempatan peluang kerja bagi masyarakat (Setiadi, 2004). Berdasarkan publikasi yang dikeluarkan oleh Badan Pusat Statistik tahun 2018 tentang Distribusi Perdagangan Komoditas Cabai Merah Indonesia Tahun 2018 menyebutkan bahwa produksi cabai besar di Jawa Timur pada tahun 2017 mencapai 100.977 ton. Sedangkan tingkat konsumsi cabai merah masyarakat Jawa Timur mencapai 3.532 ton per kapita per tahun (Malahayati, 2018).

Produktivitas cabai di Indonesia masih belum dapat dipastikan kestabilan nya dalam memenuhi kebutuhan nasional cabai masyarakat Indonesia dikarenakan produktivitas cabai yang masih fluktuatif yang disebabkan mutu benih, kualitas tanah yang kurang baik kondisi lingkungan, cuaca, penyakit dan hama yang menurunkan hasil panen ataupun menyebabkan gagal produksi (Warisno dan Dahana, 2010). Salah satu kendala yang sering dijumpai yaitu kurangnya pengetahuan para petani dalam mengenali jenis penyakit dan hama yang menyerang tanaman pada cabai (Purwanto, 2015). Sehingga kurang ada penanganan yang tepat sesuai kondisi tanaman. Hal ini di terbukti dengan adanya kasus ladang cabai di daerah Magetan bulan Juni-Juli 2019 yang terkena serangan virus. Tanaman cabai yang terkena virus memiliki lahan 0,2 hektar atau sejumlah 3.400 an batang cabai keriting. Menurut Sumarlan, koordinator POPT Kabupaten Magetan menerangkan, salah satu

penyebab dari terjadinya penyerangan virus disebabkan oleh penggunaan pestisida yang cenderung asal, dikarenakan petani rata-rata belum bisa membedakan antara hama, penyakit, atau fungi(jamur). Hal itu menyebabkan hama penyakit makin kebal (BPTP Jatim, 2019).

Clarifai adalah salah satu teknologi *Computer Vision* yang telah memenangkan 5 kali posisi teratas pada kompetisi Image Classification di *The ImageNet 2013 Competition*. *Computer Vision* adalah kemampuan untuk mengenali gambar dan vidio secara otomatis berdasarkan elemen dan pola visual. Clarifai menggunakan *Machine Learning* untuk melakukan pengenalan pada gambar dan vidio. Clarifai menggunakan Deep Convolutional Neural Network (CNNs), yaitu *subfield* dari *Machine Learning* untuk mengenali objek pada gambar dan vidio. Clarifai memiliki beberapa fitur yaitu *Image Recognition*, *Video Recognition*, *Automatic tagging*, *Content moderation*, *Visual search*, *Document classification*, *Face verification*, *Aerial surveillance*, *Predictive maintenance*, dan *Demographic analysis* (Clarifai, 2020).

Berdasarkan pemaparan masalah dan pemanfaatan teknologi yang ada, maka penulis memanfaatkan peluang untuk mengembangkan sebuah aplikasi mobile dengan sistem operasi android yang berguna untuk mendeteksi penyakit pada tanaman cabai serta pengendaliannya. Aplikasi ini berfungsi membantu petani untuk mencegah penyebaran penyakit dengan memberikan penanganan yang sesuai dengan kondisi tanaman yang terserang penyakit. Selain itu penulis juga ingin mengetahui tingkat akurasi ketepatan deteksi atau mengenali gambar dari teknologi clarifai. Berdasarkan permasalahan yang telah dijabarkan maka penulis memberi judul pada penelitian ini, yaitu “Pengembangan Aplikasi

Mobile Pendeteksi Penyakit Pada Tanaman Cabai dengan Menggunakan Teknologi Clarifai”.

Untuk melakukan pengembangan perangkat lunak, diperlukan sebuah metode untuk melakukan perencanaan, pengelolaan dan pengontrolan dari setiap proses yang dikerjakan. Metode pengembangan perangkat lunak pada aplikasi pendeteksi penyakit pada tanaman cabai menggunakan metode *Waterfall*. Metode waterfall adalah salah satu metode dari banyak nya metode pengembangan perangkat lunak atau yang biasa disebut SDLC (*Software Development Life Cycle*) yang terdiri dari lima tahapan (fase) secara berurutan yaitu bisnis analisis, perancangan, implementasi, pengujian dan pemeliharaan (Bassil, 2012). Metode *Waterfall* adalah metode yang cocok digunakan untuk pengembangan perangkat lunak dengan semua kebutuhan pada sistem telah didefinisikan di awal, tidak memiliki perubahan fungsional, fungsional tidak terlalu kompleks dan setiap tahapan dilakukan secara berurutan (Pressman, 2010).

Dari permasalahan yang telah dipaparkan, maka didapatkan beberapa rumusan masalah yaitu bagaimana hasil analisis kebutuhan fungsional dan non-fungsional pada aplikasi pendeteksi penyakit tanaman cabai?, bagaimana rancangan dan implementasi aplikasi pendeteksi penyakit pada tanaman cabai dalam aplikasi mobile dengan mengintegrasikan teknologi clarifai?, bagaimana hasil uji validasi, akurasi, *usability* dan *compatibility* pada aplikasi pendeteksi penyakit pada tanaman cabai?.

2. LANDASAN KEPUSTAKAAN

2.1. Penelitian Terkait

Terdapat beberapa penelitian yang relevan dengan penelitian ini, sehingga dapat membantu dalam pengerjaan.

Penelitian pertama ialah Sistem Pakar Analisa Penyakit Pada Tanaman Cabai Merah Menggunakan Metode Backward Chaining (Nusantara, Pamungkas, Syaifudin, Kusuma, & Fikri, 2017). Pada penelitian ini, terdapat masalah yaitu kurangnya pemahaman petani dalam menanggulangi penyakit pada cabai merah yang diatasi dengan sebuah solusi yaitu membuat sistem informasi berbasis web menggunakan metode backward Chaining untuk membantu para petani dalam menganalisis penyakit pada tanaman cabai. Hasil dari

penelitian ini yaitu sebuah produk berbasis web yang dapat membantu para petani untuk mendeteksi penyakit pada tanaman cabai akan tetapi dengan menggunakan backward chaining masih terdapat beberapa kekurangan dalam menentukan pola solusi.

Penelitian kedua ialah Sistem pakar Pakar Deteksi Hama dan Penyakit Pada Tanaman Cabai Dengan Metode Naïve Bayes (Fistrianingtyas & Rahmad, 2015). Pada penelitian ini, terdapat masalah yaitu keterbatasan jumlah pakar atau ahli pertanian tidak dapat mengatasi permasalahan petani cabai yang diatasi dengan membuat sebuah sistem pakar untuk mendeteksi hama dan penyakit pada tanaman cabai menggunakan metode naïve bayes dalam proses identifikasi dengan media web. Hasil dari penelitian ini adalah dapat mengidentifikasi penyakit berdasarkan banyaknya data kejadian yang telah dimasukkan oleh pakar.

Penelitian ketiga ialah Rancang Bangun Aplikasi *SmartFoodies* Dengan Memanfaatkan Clarifai Api Untuk *Image Recognition* Berbasis Android (Ryantono, 2017). Pada penelitian ini terdapat permasalahan yaitu kesulitan masyarakat untuk mengenal dan membuat berbagai macam makanan khas nusantara yang harus dilestarikan. Solusi yang diberikan oleh peneliti yaitu membangun aplikasi *smartfoodies* yaitu aplikasi untuk mempermudah pengguna dalam mengetahui nama bahan dan informasi pada makanan dengan akurat menggunakan teknologi clarifai untuk melakukan *image recognition* berbasis android. Hasil dari penelitian ini yaitu membantu para pengguna dalam mengetahui tata cara masak, pembuatan resep masakan berdasarkan pemanfaatan bahan yang ada dan menentukan rekomendasi resep makanan.

Penelitian terakhir yaitu *Agile vs waterfall: A Comparative Analysis*. Pada penelitian ini dilakukan perbandingan antara dua buah metode pengembangan perangkat lunak yaitu metode pengembangan *Agile* dan *Waterfall* sebagai landasan dalam pemilihan metode untuk pengembangan perangkat lunak. Pada penelitian tersebut dapat disimpulkan bahwa untuk mengembangkan perangkat lunak yang berbeda maka harus menggunakan metode yang berbeda pula tergantung tingkat kompleksitas dan daur hidup perangkat lunak yang dibangun. Jika suatu perangkat lunak yang akan dibangun memiliki kebutuhan yang tidak berubah maka metode yang paling sesuai yaitu menggunakan metode

Waterfall, namun jika kebutuhan fungsional yang memungkinkan untuk berubah saat proses pengerjaan maka metode yang paling sesuai adalah menggunakan metode *Agile* (Kannan, Smita, & Verma, 2014).

2.2. Penyakit pada Tanaman Cabai

2.2.1 Layu Fusarium(*Fusarium oxysporum f. Sp*)

Penyakit layu fusarium pada tanaman cabai disebabkan oleh cendawan *fusarium oxysporum*. Gejala yang dapat terlihat pada tanaman cabai yang terkena penyakit ini yaitu, tanaman mulai mengalami kelayuan dari bawah dan menguning menjalar ke atas ranting muda. Sumber penyakit ini biasanya berasal dari tanah dan sisa tanaman sakit. Adapun pemicu perkembangan penyakit layu fusarium yaitu lahan berpasir, pupuk N(ZA) terlalu tinggi, kurangnya pupuk kandang, tanah kekurangan kalsium dan jumlah nematoda tinggi. Penyakit layu fusarium pada tanaman cabai dapat dilihat pada Gambar 1. **Error! Reference source not found.**



Gambar 1 Layu Fusarium

2.2.2 Virus Kuning(*Gemini Virus*)

Pada Gambar 2, menunjukkan gejala penyakit virus kuning. Penyebab penyakit virus kuning pada tanaman cabai yaitu *gemini virus*. Gejala yang dapat dilihat dari tanaman cabai yang terkena penyakit virus kuning adalah warna kuning pada daun yang terlihat jelas dan tulang daun berubah menjadi kuning terang dan menebal serta daun yang menggulung ke atas.



Gambar 2 Virus Kuning

2.2.3 Virus Kuning(*Gemini Virus*)

Pada Gambar 3, menunjukkan gejala bercak daun. Penyakit bercak daun pada tanaman cabai disebabkan oleh *Cercospora capsici* Heald and Wolf. Bercak daun *cercospora* dapat menimbulkan defoliasi jika serangan terjadi pada daun, sedangkan apabila terjadi pada bunga akan mengakibatkan gugur bunga serta apabila terjadi pada buah maka dapat menimbulkan mal formasi pada buah yang mengakibatkan buah menjadi kerdil. Gejala penyakit ini menimbulkan munculnya bercak bulat berwarna cokelat pada daun dengan kondisi yang kering serta memiliki ukuran sekitar 1 inci. Bercak yang tua dapat menyebabkan lubang-lubang pada bagian daun.

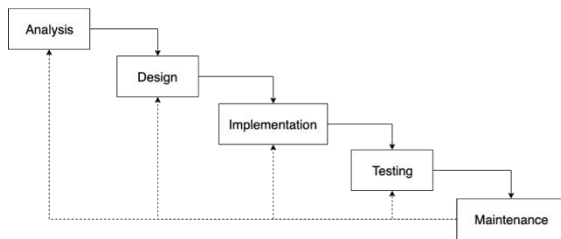


Gambar 3 Bercak daun

2.3. Waterfall

Dalam pengembangan aplikasi penyakit pada tanam cabai ini peneliti menggunakan salah satu model pengembangan perangkat lunak yaitu Waterfall. Waterfall merupakan salah satu dari siklus pengembangan perangkat lunak atau yang biasa di sebut dengan SDLC (*Software Development Life Cycle*). Pada **Error! Reference source not found.**, menunjukkan gambar dari metode *waterfall* yang dibagi kedalam fase-fase dari setiap proses

pengembangan yang mirip dengan air terjun. Cara kerja waterfall yaitu secara berurutan dengan lima fase yang harus dilalui untuk mengembangkan suatu perangkat lunak (Bassil, 2012). Siklus pengembangan *waterfall* ditunjukkan pada Gambar 4.



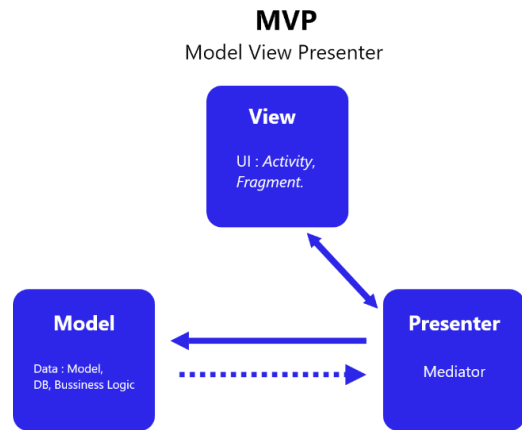
Gambar 4 Siklus Pengembangan Waterfall

2.4. Best Practice

Dalam membangun aplikasi perangkat bergerak, diperlukan sebuah arsitektur perangkat lunak untuk memudahkan pengembang dalam membangun aplikasi tersebut.

2.4.1. Model View Presenter (MVP)

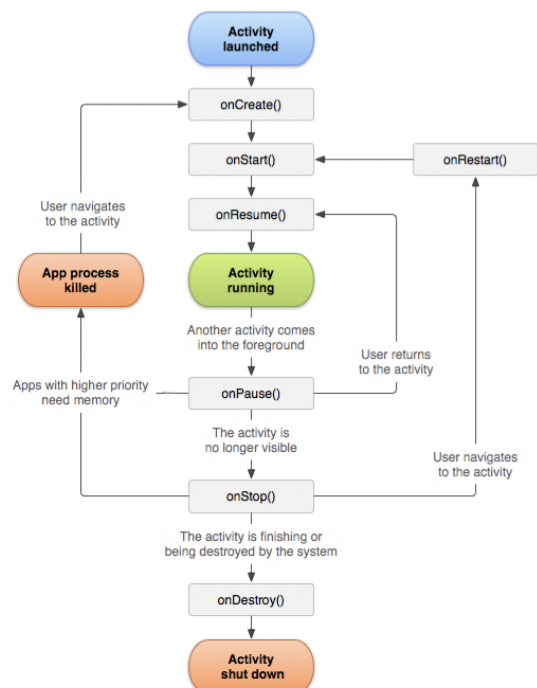
Permasalahan yang sering muncul dalam mengembangkan perangkat lunak aplikasi adalah sulitnya melakukan *testing* (pengujian) dan *maintenance* (pemeliharaan) dikarenakan kode yang dibuat tidak terstruktur dan terkelompokkan dengan baik. Maka aplikasi yang dibangun harus memiliki arsitektur atau bagian-bagian (*layer*) yang terpisah sesuai dengan perannya masing-masing. Salah satu arsitektur dalam mengembangkan perangkat lunak aplikasi adalah MVP (Model View Presenter). Sesuai dengan namanya MVP memiliki 3 bagian, yaitu Model, View dan Presenter. *Model* berfungsi untuk mengurus bagian dari bisnis data, *View* adalah bagian yang menangani semua bagian UI (*User Interface*) dan *Presenter* adalah bagian yang berfungsi untuk menghubungkan antara *View* dan *Model*. Arsitektur MVP merupakan salah satu arsitektur yang dapat digunakan oleh pengembang yang memiliki keterbatasan dalam kemampuan pemrograman yang terbatas (Ojeda-Guerra, 2015). Gambaran MVP ditunjukkan pada Gambar 5.



Gambar 5 MVP Architecture

2.5 Android

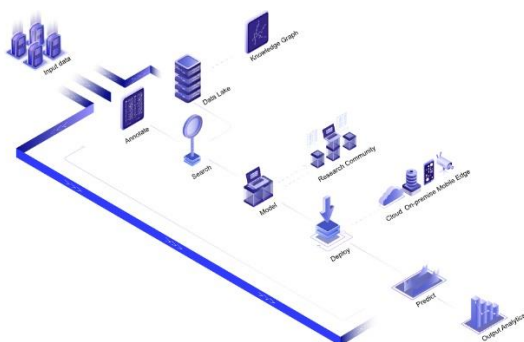
Android adalah sistem operasi berbasis kernel Linux yang dirancang oleh Google yang biasa digunakan untuk perangkat seperti *smartphone*, *tablet*, *smartwatch* dan berbagai *smart device* lainnya. Android juga memiliki SDK (Software Development Kit) yang membantu dan mempermudah *developer* untuk mengembangkan aplikasi. Android juga memiliki banyak versi yang sangat beraneka ragam seperti: *Cupcake*, *Donut*, *Éclair*, *Froyo*, *Gingerbread*, *Honeycomb*, *Ice Cream Sandwich*, *Jelly Bean*, *Kit Kat*, *Lollipop*, *Marshmallow*, *Nougat*, *Oreo*, *Pie* dan versi android yang baru di rilis adalah Android 10 (Developers, 2019). Pada Gambar 6, *flow* gambaran dari life cycle android.



Gambar 6 Android Activity Lifecycle

2.6 Clarifai

Clarifai adalah perusahaan *Artificial Intelligence* yang bergerak di bidang *Computer Vision* menggunakan *Machine Learning* dan *Neural Network* untuk mengidentifikasi gambar baik berupa photo ataupun video. Clarifai memiliki API yang dapat digunakan dalam mengidentifikasi dan mengklasifikasi citra atau gambar secara *custom*. Selain itu clarifai juga menyediakan SDK untuk android maupun iOS untuk membantu *developer* mengembangkan aplikasi dengan kemampuan seperti image classification, object detection dan lain-lain (Clarifai, 2019). Clarifai platform diagram ditunjukkan pada Gambar 7.



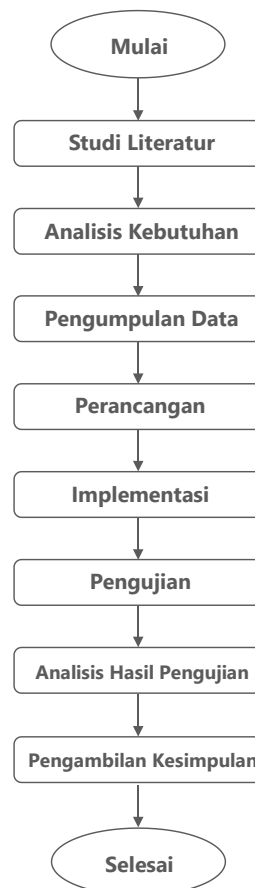
Gambar 7 Clarifai Platform Diagram

2.6 Imgur

Imgur adalah merupakan salah satu website hosting gambar dan situs diskusi komunitas (imgur, 2020). Imgur memiliki fitur seperti media sosial instagram maupun facebook dimana setiap pengguna dapat mencari, mengunggah dan mengomentari gambar yang ada pada situs tersebut. Imgur dapat dimanfaatkan oleh developer untuk membantu melakukan *hosting image* melalui API yang disediakan oleh imgur. API yang digunakan memberikan akses untuk dapat menyimpan gambar di database imgur sehingga pengembang dapat mengakses gambar tersebut melalui URL yang didapatkan dari imgur.

3. METODOLOGI

Untuk melakukan pengembangan sebuah perangkat lunak, maka dibutuhkan suatu metode untuk agar proses pengembangan dapat dilakukan secara terstruktur dengan baik. Adapun alur diagram dari penelitian ini ditunjukkan pada Gambar 8.



Gambar 8 Diagram Alur Penelitian

4. ANALISIS KEBUTUHAN

Pada tahapan ini segala kebutuhan dalam sistem akan dijelaskan secara umum dan lengkap. Penentuan lingkungan sistem yang akan dibuat yang kemudian menentukan calon pengguna yang akan berperan sebagai aktor pada aplikasi ini. Setelah itu fungsionalitas akan ditetapkan berdasarkan hasil wawancara dan observasi. Setelah fungsional didapatkan maka akan digambarkan kedalam *use case diagram*.

Pada tahap analisis kebutuhan dalam sistem ini akan dijelaskan secara umum. Penentuan calon pengguna dibuat berdasarkan lingkungan pada sistem yang akan berperan sebagai aktor. Fungsionalitas didapatkan berdasarkan wawancara dan observasi yang selanjutnya akan digambarkan kedalam *use case diagram*.

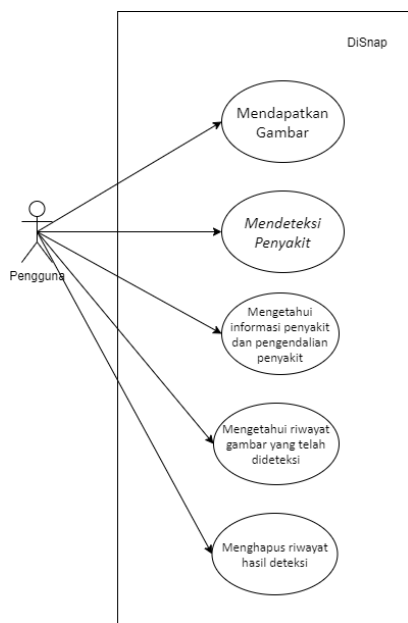
Berdasarkan hasil dari dilakukannya analisis kebutuhan sistem didapatkan hasil dari sebuah identifikasi aktor yang dapat dilihat pada Tabel 1.

Tabel 1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna adalah aktor yang menggunakan seluruh fitur pada sistem. Pengguna berinteraksi dengan sistem secara langsung

untuk melakukan proses mendeteksi penyakit pada tanaman cabai. Pengguna aplikasi dapat seorang petani, pakar ataupun peneliti tanaman cabai.

Selanjutnya didapatkan *use case diagram* dari sistem. *Use case diagram* aplikasi pendeteksi penyakit pada tanaman cabai dapat dilihat pada Gambar 9.



Gambar 9 Use case Diagram

5. PERANCANGAN

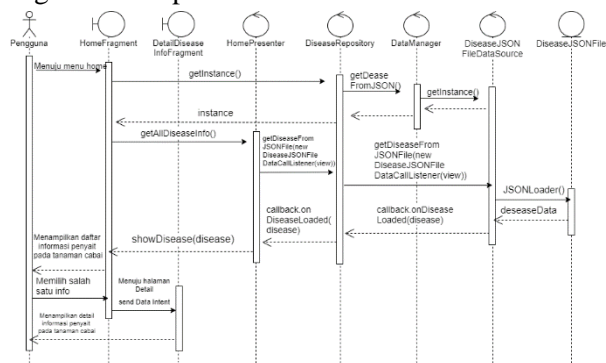
Tahap perancangan aplikasi DiSnap terdiri dari perancangan arsitektur sistem, sequence diagram, class diagram, perancangan basis data, perancangan antarmuka pengguna (*wireframe*) dan perancangan algoritme.

Perancangan arsitektur sistem dibuat untuk memberikan gambaran umum terhadap sistem yang akan dibangun. Gambaran umum sistem ini dapat membantu mempermudah untuk memahami alur kerja pada sistem yang akan dibangun. Layanan yang digunakan pada aplikasi DiSnap yaitu *imgur* dan *clarifai*. *Imgur* berfungsi untuk melakukan *image hosting* gambar yang diambil oleh pengguna melalui galeri atau kamera. Sedangkan *clarifai* digunakan untuk mengenali nama penyakit dari gambar yang telah di ambil oleh pengguna. Arsitektur kode yang digunakan dalam pengembangan aplikasi ini adalah MVP (*Model View Presenter*).

Sequence diagram merupakan diagram yang menggambarkan aktivitas dan interaksi antar komponen secara berurutan. Semua komponen yang ada pada *sequence diagram*

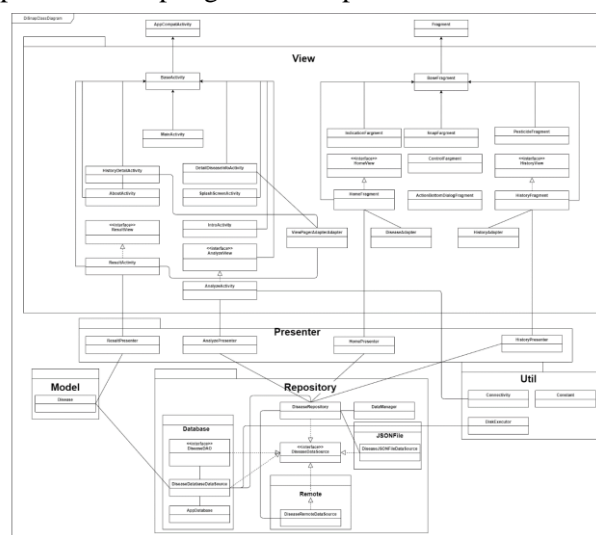
merupakan hasil dari analisis dan identifikasi dari kebutuhan serta skenario *use case* yang ada pada tahap analisis kebutuhan sebelumnya. Berikut salah satu contoh sequence diagram yang ada pada perancangan sistem seperti pada

Sequence diagram menjelaskan interaksi antar class pada sistem secara berurutan sehingga dapat mencapai setiap tujuan dari kebutuhan fungsional. Contoh dalam perancangan sistem dengan *sequence diagram* digambarkan pada Gambar 10.



Gambar 10 Sequence Diagram Mengetahui Informasi Penyakit dan Pengendalian Penyakit

Selain *sequence diagram*, dalam melakukan perancangan juga dibuat *class diagram* yang berguna untuk mengetahui hubungan antar *class*, sehingga dapat menjadi rujukan dalam melakukan proses pengembangan. Contoh class diagram pada aplikasi DiSnap digambarkan pada Gambar 11.



Gambar 11 Class Diagram DiSnap

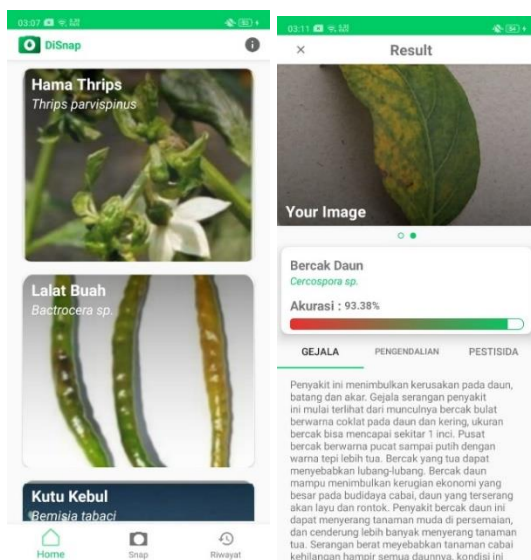
Pada bagian akhir perancangan, dilakukan perancangan antar muka dengan menggunakan *wireframe*. Contoh warframe dalam aplikasi DiSnap ditunjukkan pada Gambar 12.



Gambar 12 Waframe Home

6. IMPLEMENTASI

Pada tahap implementasi peneliti mulai melakukan pengembangan aplikasi DiSnap dengan berpedoman pada hasil perancangan yang telah dibuat. Dalam proses model *Waterfall*, maka peneliti sudah memasuki tahap Implementasi. Pembahasan yang terdapat pada implementasi yaitu terdiri dari spesifikasi sistem, batasan implementasi, implementasi kode program, dan implementasi antarmuka. Salah contoh implementasi antarmuka aplikasi DiSnap ditunjukkan pada Gambar 13.



Gambar 13 Implementasi Aplikasi DiSnap

7. PENGUJIAN

Pengujian ini dilakukan untuk mengetahui

apakah aplikasi dapat berfungsi sesuai kebutuhan pengguna dan memastikan bahwa aplikasi dapat berjalan dengan baik dan benar. Pengujian yang akan dilakukan pada sistem yaitu pengujian validasi (*blackbox testing*), pengujian akurasi, pengujian kompatibilitas (*compatibility testing*), dan pengujian *usability* (*usability testing*). Selanjutnya pada setiap pengujian akan dibuat hasil analisis.

Pengujian validasi merupakan pengujian yang bertujuan untuk memastikan fungsionalitas yang terdapat pada sistem bekerja sesuai dengan kebutuhan fungsionalitas pengguna melalui skenario *use case* yang telah dibuat sebelumnya. Hasil pengujian validasi pada aplikasi DiSnap bernilai valid untuk semua kasus uji. Artinya aplikasi sudah dapat berfungsi dengan baik sesuai kebutuhan yang telah didefinisikan. Dengan begitu aplikasi DiSnap dapat dikatakan berhasil dengan tingkat keberhasilan 100% sesuai fungsionalitas yang ada.

Pengujian akurasi adalah salah satu jenis pengujian yang bertujuan untuk mendapatkan nilai ketepatan aplikasi DiSnap untuk mendeteksi jenis penyakit pada tanaman cabai melalui daun ataupun buah yang terkena penyakit. Hasil pengujian akurasi pada aplikasi DiSnap yaitu mendeteksi gambar daun yang terkena penyakit pada tanaman cabai dengan menggunakan clarifai, didapatkan persentase akurasi dari pengujian data uji sebesar 70.59%. Dilihat dari hasil persentase nya maka keakuratan dari aplikasi ini dapat dikatakan belum terlalu akurat, hal ini disebabkan karena kurang banyaknya data latih yang digunakan dalam proses data latih pada website clarifai serta kualitas gambar yang dideteksi.

Pengujian *usability* dilakukan dengan tujuan untuk mengetahui tingkat kegunaan terhadap pengguna dari aplikasi yang telah dikembangkan. Pengujian *usability* dilakukan terhadap 5 responden pegawai BPTP Jawa Timur seperti pakar penyakit hama dan tanaman, petani dan pekerja BPTP Jawa Timur lainnya. Hasil dari pengujian *usability* pada aplikasi DiSnap menggunakan SUPR-Qm didapatkan nilai *usability* SUPR-Qm yaitu 82.75%. Nilai tersebut di konversikan kedalam skala konversi kategori nilai *usability* maka akan masuk kedalam kategori nilai B atau *Excellent*. Sehingga dapat disimpulkan aplikasi DiSnap dinilai baik dan dapat diterima oleh para pakar dan petani tanaman cabai di Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur.

Pengujian *compatibility* yang dilakukan pada penelitian ini memiliki tujuan untuk mengetahui apakah aplikasi yang sudah dikembangkan dapat berjalan atau beroperasi pada perangkat dengan sistem operasi Android yang sudah ditentukan. Pada penelitian aplikasi yang dikembangkan sudah di atur untuk dapat dioperasikan dengan sistem operasi Android minimal level 23 dengan target level 29. Artinya aplikasi hanya akan berjalan pada perangkat dengan minimal level 23. Dengan menggunakan bantuan dari firebase test lab untuk melakukan pengujian *compatibility*, didapatkan hasil pengujian yaitu bahwa aplikasi DiSnap hanya dapat berjalan pada aplikasi android dengan minimal SDK level 23 (Marshmallow).

8. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi dan pengujian pada pengembangan aplikasi pendeteksi penyakit pada tanaman cabai menggunakan teknologi clarifai, maka didapatkan 3 kesimpulan sebagai berikut:

Hasil analisis kebutuhan pada pengembangan aplikasi pendeteksi penyakit tanaman cabai didapatkan dengan cara penggalan kebutuhan menggunakan teknik wawancara terhadap seorang pakar penyakit dari Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur. Selain itu dilakukan juga observasi ke lapangan yaitu ke lahan pertanian cabai di Malang yang berada di bawah naungan BPTP Jawa Timur. Selanjutnya dilakukan proses identifikasi aktor, kebutuhan sistem dan pemodelan kebutuhan pada sistem. Aplikasi mobile pendeteksi penyakit pada tanaman cabai memiliki 5 kebutuhan fungsional dan 2 kebutuhan non fungsional.

Perancangan yang dilakukan pada aplikasi pendeteksi penyakit pada tanaman cabai menghasilkan perancangan arsitektur sistem, *sequence diagram*, *class diagram*, ERD (*Entity Relationship Diagram*), *wireframe* (perancangan antarmuka) dan perancangan algoritme. Implementasi aplikasi pendeteksi penyakit pada tanaman dilakukan pada sistem operasi android dengan memanfaatkan teknologi clarifai untuk melakukan pengenalan penyakit pada tanaman cabai melalui gambar daun cabai yang diambil oleh pengguna menggunakan kamera atau galeri dan diunggah ke imgur secara otomatis. Aplikasi pendeteksi penyakit pada tanaman cabai

diimplementasikan dengan bahasa pemrograman Java dan menggunakan arsitektur MVP (*Model View Presenter*). Integrasi teknologi clarifai pada pengembangan aplikasi mobile pendeteksi penyakit pada tanaman cabai diperlukan 3 cara yaitu *define*, *train* dan *recognize*. Pada tahap *define* dilakukan pengelompokan dan pelabelan gambar menjadi 6 label yaitu Bercak Daun, Busuk Buah Antraknosa, Busuk Daun, Hama Thrips, Hama Tungau, dan Virus Kuning. Pada tahap *train* dilakukan pelatihan oleh clarifai untuk dapat mengenali penyakit pada tanaman cabai. Pada tahap *recognize* dilakukan pengenalan pada gambar yang dikirimkan kepada website clarifai yang sebelumnya telah dilakukan *hosting image* pada website imgur menggunakan *web service API* yang disediakan sehingga didapatkan nama penyakit dengan tingkat akurasi dari ketepatan pendeteksian penyakit dalam bentuk JSON. Data tersebut diolah pada perangkat pengguna dan ditampilkan nama penyakit, nama latin penyakit, detail penyakit serta tingkat akurasi dari ketepatan teknologi clarifai dalam mendeteksi penyakit.

Pengujian validasi yang telah dilakukan pada sistem yaitu dengan menguji fungsional sistem menghasilkan bahwa aplikasi pendeteksi penyakit pada tanaman cabai memiliki tingkat 100% pada fungsionalitas dikarenakan semua hasil pengujian validasi bernilai valid. Pada pengujian akurasi mendeteksi penyakit cabai menggunakan teknologi clarifai didapatkan akurasi sebesar 70.59% berdasarkan pengujian pada data uji gambar sebanyak 100 gambar. Akurasi dari aplikasi ini dapat dikatakan belum akurat karena belum semua data uji yang diuji memiliki ketepatan dalam mendeteksi ketepatan nama penyakit yang ada pada gambar karena kurangnya data latih dan kualitas gambar. Sehingga dapat disimpulkan bahwa aplikasi ini dapat menghasilkan akurasi yang baik apabila memiliki banyak data latih dan kualitas gambar yang baik. Pada pengujian *usability* didapatkan skor dengan nilai 82.75%. Nilai tersebut termasuk kedalam kategori nilai B atau *Excellent*. Sehingga dapat disimpulkan aplikasi DiSnap dinilai baik dan dapat diterima oleh para pakar dan petani tanaman cabai di Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur. Sedangkan untuk pengujian *compatibility*, berdasarkan pengujian menggunakan Firebase Test Lab didapatkan hasil bahwa sistem dapat beroperasi sesuai

dengan minimal level SDK Android yang telah ditentukan oleh peneliti yaitu level 23 (Android Marshmallow).

9. DAFTAR PUSTAKA

- Ahmad, A. 2017. Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Yayasan Cahaya Islam, Jurnal Teknologi Indonesia*.
- Akbari, G. W., Hidayat, N. & Santoso, N., 2019. Diagnosis Penyakit Cabai Menggunakan Metode Fuzzy K-Nearest Neighbor (FKNN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 03(1), 1070-1074.
- Bassil, Y., 2012. A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (IJET)*, 02(05), 2049–3444.
- Brooke, J. (1986). SUS - A quick and dirty usability scale. Usability Evaluation in Industry. Retrieved from <https://cui.unige.ch/isi/icle-wiki/media/ipm:testsuschart.pdf>.
- Brooke, J. (2013). SUS: A Retrospective. *Journal of Usability Studies*, 8(2), 29–40.
- Clarifai. 2019. About the platform. [Online] Tersedia <https://www.clarifai.com/about> [Diakses 16 Juli 2019].
- Darmawan, D.A. and E. Pasandaran. 2000. Indonesia. In: M. Ali (ed). *Dynamic of vegetable production. distribution and consumption in Asia*. AVRDC Publication 00-498. Shanhu. Tainan: AVRDC. Pp.139-171. <http://www.avrdc.org/pdf/dynamics/Indonesia.pfd>.
- Developers, G., 2018. *About the platform*. [Online] Tersedia di: <https://developer.android.com/about/> [Diakses 16 Juli 2019].
- Developers, A. (2019c). Save data in a local database using Room. Retrieved July 24, 2019, from <https://developer.android.com/training/datastorage/room/index.html>.
- Firebase. (2019). Firebase Test Lab Robo tests. Retrieved October 1, 2019, from <https://firebase.google.com/docs/test-lab/android/robo-ux-test>.
- Imgur. *About Imgur*. [Online] Tersedia di: <https://imgurinc.com/> [Diakses 30 April 2020].
- JSON. (2019). Introducing JSON. Retrieved September 15, 2019, from <https://www.json.org/> [Diakses 16 Juli 2019].
- Kannan, V., Smita, J., & Verma, S., 2014. *Agile vs waterfall: A Comparative Analysis*. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 3(10), 2680–2686.
- Henriyadi, H., & Mulyati, R. (2016). USABILITY TESTING Sistem Informasi: Studi kasus pada Aplikasi Repositori Publikasi Badan Penelitian dan Pengembangan Pertanian. *Jurnal Perpustakaan Pertanian*, 23(2), 54. <https://doi.org/10.21082/jpp.v23n2.2014.p54-63>.
- Henstam, P., 2018. *How many participants are needed when usability testing physical products?: An analysis of data collected from usability tests conducted on physical products*.
- Jatim, BPTP. (2019, 6 Juli). Beredar Kabar Ladang Cabai di Magetan Diserang Virus, Ini Faktanya. [Online] Tersedia di <http://jatim.litbang.pertanian.go.id/beredar-kabar-ladang-cabai-di-magetan-diserang-virus-ini-faktanya/> [Diakses 16 Juli 2019].
- Malahayati, Nur., Fadhli, Muhammad. 2018. *Distribusi Perdagangan Komoditas Cabai Merah Indonesia Tahun 2018*. Jakarta: BPS RI.
- MeasuringU. (n.d.). SUPR-Q Full License. Retrieved April 2, 2020, from <https://measuringu.com/product/suprq/>.
- Meilin, A. 2014. *Hama dan Penyakit pada Tanaman Cabai Serta Pengendaliannya*. 2014. Jambi: Balai Pengkajian Teknologi Pertanian (BPTP) Jambi.
- Menanti, S. 2019. Analisis Perkembangan Harga Bahan Pangan Pokok di Pasar Domestik dan Internasional. Jakarta: Kementerian Perdagangan Indonesia.
- Nusantara, D.M., Pamungkas, S. W., Syaifudin, N. R., Kusuma, L. W., Fikri, J. 2017. Sistem Pakar Analisa Penyakit Pada Tanaman Cabai Merah Menggunakan Metode Backward Chaining. *Seminar*

- Nasional Teknologi Informasi dan Multimedia*, 2302 - 3805.
- Ojeda-Guerra, C. N., 2015. *A Simple Software Development Methodology Based on MVP for Android Applications in a Classroom Context*. 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 1429–1434.
- Pressman, Roger S. 2010. *Software Engineering: A Practitioner's Approach, Seventh Edition*. New York: McGraw-Hill.
- Purwanto, T. & Destiani, D. 2015. Pengembangan Sistem Pakar Diagnosis Penyakit Cabai. Garut: Jurnal STT-Garut.
- Ryantono, R. P., 2017. Rancang Bangun Aplikasi Smartfoodies Dengan Memanfaatkan Clarifai Api Untuk Image Recognition Berbasis Android.
- Sauro, J., & Lewis, J. R., 2012. *Quantifying the User Experience: Practical Statistics for User Research* (1st ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Sauro, J., & Zarolia, P. (2017). SUPR-Qm: A Questionnaire to Measure the Mobile App User Experience. *Journal of Usability Studies*, 13(1), 17–37.
- Setiadi. 2004. *Bertanam Cabai*. Penebar Swadaya. Jakarta. 12 hlm. Warisno dan Dahana. 2010. *Peluang Usaha dan Budidaya Cabai*. Jakarta: Gramedia Pustaka Utama.
- Setiadi. 2011. *Bertanam Cabai di Lahan Pot*. Jakarta: Penebar Swadaya.
- Statcounter, 2019. *Operating System Market Share in Indonesia*. [Online] Tersedia di : <https://gs.statcounter.com/os-market-share/all/indonesia> [Diakses 26 Agustus 2019].
- Warisno dan Dahana. 2010. *Peluang Usaha dan Budidaya Cabai*. Jakarta: Gramedia Pustaka Utama.
- Wu, B., & Yang, X., 2009. *Application of MVP Architecture in Reengineering of Legacy Financial System*. 2009 International Conference on Computational Intelligence and Software Engineering, 1–5.
- Zhang, T., Gao, J., Cheng, J., & Uehara, T. (2015). Compatibility testing service for mobile applications. *Proceedings - 9th IEEE International Symposium on Service-Oriented System Engineering, IEEE SOSE 2015*, 30(April), 179–186. <https://doi.org/10.1109/SOSE.2015.35>