

BAB IV HASIL DAN PEMBAHASAN

Berisi tentang proses pembuatan, pengujian dan hasil dari aplikasi dengan mengimplementasikan algoritma *genetika*.

BAB V SIMPULAN DAN SARAN

Memuat simpulan yang didapat dalam membuat aplikasi serta memuat saran yang terkait dengan penelitian.

BAB II

TINJAUAN PUSTAKA

2.1 Aplikasi Tani Cerdas

Tani Cerdas adalah aplikasi yang di kembangkan oleh mahasiswa dan dosen Teknik Informatika IIB Darmajaya dengan tujuan mempermudah pengecekan tanaman, pada saat ini aplikasi Tani Cerdas hanya memiliki fitur monitoring kelembaban tanah, keasaman tanah, kelembaban udara, dan suhu udara yang berfungsi untuk penyiraman berbasis fuzzy logic dan pemberian pupuk (Syahputri et al., 2022).

Aplikasi Tani Cerdas akan terus dikembangkan agar dapat menjadi aplikasi yang dapat dipergunakan luas oleh petani Indonesia salah satunya pengembangan fitur To Do List budidaya tanaman melon buah melon bertujuan menjadi panduan atau rujukan dalam budidaya melon tampilan aplikasi bisa dilihat pada gambar 2.1 dan gambar 2.2.



*Gambar 2.1 Tampilan Splash Screen
Gambar Tani Cerdas*



*Gambar 2.2 Tampilan Menu
Pada Aplikasi Tani Cerdas*

2.2 Android Studio

Android Studio adalah lingkungan pengembangan terpadu (Integrated Development Environment/IDE) resmi untuk pengembangan aplikasi android, yang didasarkan pada IntelliJ IDEA (*Android Developers*, n.d.) . Selain sebagai editor kode dan alat pengembang IntelliJ, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas tenaga kerja saat membangun aplikasi android. Android Studio menyediakan alat pengembangan android terintegrasi untuk mengembangkan dan men-debug program. IntelliJ IDEA sendiri adalah Java IDE komersial yang kuat seperti Eclipse dan Netbeans yang dikembangkan oleh JetBrains.

2.3 To Do List

To Do List adalah sebuah dasar dari manajemen waktu yang paling mudah dan sederhana. Walaupun paling mudah, mendasar, dan sederhana, jika diterapkan secara konsisten maka dapat mendongkrak produktivitas hidup secara signifikan (Oktarina & Hajjah, 2019).

2.4 MySQL

MySQL adalah perangkat lunak RDBMS (atau server database) yang dapat melakukan itu kelola database dengan sangat cepat dan serap data dalam jumlah besar. Perangkat lunak ini dapat diakses oleh banyak pengguna (multiuser), memiliki kekuatan pemrosesan sinkron atau bersamaan (multihead), dan mengeksekusi proses secara sinkron atau bersamaan (multithreaded). MySQL sudah banyak digunakan saat ini oleh berbagai kalangan. Penyimpanan, manajemen data, industri penyimpanan dan manajemen data sains, usaha menengah, industri besar. MySQL adalah salah satu aplikasi DBMS yang paling banyak digunakan oleh pemrogram aplikasi web. Kelebihan MySQL adalah gratis, selalu diperbarui, dan berguna ketika pengguna mengalami masalah (Sitinjak Daniel Dido Jantce TJ & Suwita, 2020).

2.5 Adobe XD

Program ini dikenal sebagai Adobe Experience Design atau Adobe XD. Tidak diragukan lagi, setiap paket software Adobe Design selalu

memiliki kelebihan masing-masing. Adobe XD adalah perangkat lunak desain berbasis vektor untuk aplikasi seluler dan web. Adobe XD secara resmi dirilis pada 14 Maret 2016, menghadirkan fitur dan alat responsif yang lebih terkenal. Kehadiran Adobe XD memungkinkan desainer aplikasi seluler untuk bekerja tanpa masalah (Pamungkas, 2016). Perangkat lunak ini mendukung pembingkai kawat dan pembuatan prototipe situs web. Adobe menawarkan layanan gratis dalam Adobe XD untuk membangun UI, UX, dan pembuatan prototipe di berbagai platform seperti web, seluler, dan tablet. Sistem operasi yang didukung termasuk Windows dan Mac OS, dan versi terbaru juga tidak terkunci di android dan ios.

2.6 Java

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada computer ataupun pada lingkungan jaringan (DeCoster, 2012) . Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari sun microsystem yang memiliki kode green dengan tujuannya adalah untuk membuat bahasa komputer sederhana yang dapat berjalan pada perangkat sederhana tanpa terikat pada arsitektur tertentu, dan pada awalnya disebut Oak, itu sendiri merupakan nama untuk sebuah bahasa pemrograman komputer yang ada. maka sun mengubahnya menjadi java (Mauluddin et al., 2018) . Sun kemudian meluncurkan browser Java bernama Hot Java yang dapat menjalankan applet. Kemudian Netscape mengadopsi teknologi Java, memungkinkan program Java untuk berjalan di browser Netscape, diikuti oleh Internet Explorer. Teknologi Java mulai diminati banyak vendor seperti IBM, Symantec, dan Inprise karena keunikan dan keunggulannya.

2.7 Penjadwalan

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan di mana setiap operasi sebagai bagian dari pekerjaan secara

keseluruhan harus dilakukan pada sumber daya yang terbatas. Penjadwalan adalah proses untuk melakukan tugas dengan menggunakan sumber-sumber yang tersedia pada waktu yang telah ditetapkan (Pamungkas, 2016). setiap waktu tertentu, biasanya memberitahukan hal-hal yang seharusnya terjadi dan menunjukkan kepada rencana untuk menentukan waktu suatu kegiatan. Jadwal dapat dengan awal atau penyelesaian dari sebuah pekerjaan. Untuk perencanaan memutuskan waktu dalam kegiatan. Jadwal dapat dengan dimulainya atau selesainya suatu pekerjaan. Perencanaan adalah langkah terakhir dalam proses perencanaan, Penjadwalan yang baik tentukan produktivitas tenaga kerja saat bekerja karena dapat memutuskan di mana pekerja harus bekerja, Istirahat dan liburan untuk menjaga performa energi dan kesehatan dalam bekerja.

Penjadwalan merupakan sebuah rencana dibuat untuk mengelola sumber daya yang ada. Untuk memenuhi sumber daya yang terbatas seperti kapasitas dan waktu. Menghindari penggunaan waktu dan sumber daya yang berlebihan. Oleh karena itu, sangat penting untuk memahami konsep penjadwalan tenaga kerja (Ilmi et al., 2015). Dengan demikian, tenaga kerja dapat mengetahui kapan harus memulai dan menyelesaikan pekerjaan waktu kerja. Perencanaan tenaga kerja yang baik ini memiliki efek positif dari efisiensi dan efektivitas waktu sebagai tenaga kerja.

Untuk menentukan waktu terbaik, ada tiga variable yang dipakai, yaitu curah hujan, suhu, dan kelembapan (*Penentuan Waktu Tanam Sebagai Bentuk Antisipasi / Warung Ilmiah Lapangan, n.d.*)

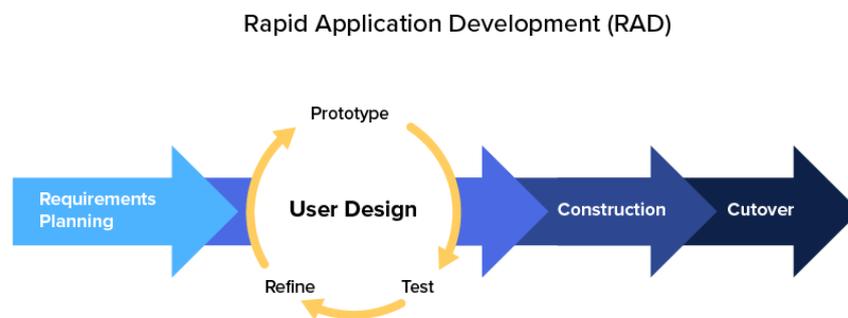
2.8 Metode Pengembang Perangkat Lunak

RAD atau *Rapid Application Development* adalah pendekatan berorientasi objek untuk pengembangan sistem, termasuk metodologi pengembangan dan perangkat lunak (Trimahardhika, 2017). Metode ini

menekankan pada proses pembuatan aplikasi berdasarkan pembuatan *prototype*, iterasi dan *feedback* yang berulang-ulang.

Metode ini berbeda dengan metode pengembangan lainnya, seperti waterfall, dinilai tidak efektif karena membutuhkan aplikasi yang dirancang dari awal hingga akhir. Hal ini mempengaruhi lama waktu yang dibutuhkan untuk merilis sebuah aplikasi (Trimahardhika, 2017).

Trimahardhika (Trimahardhika, 2017) menyampaikan bahwa metode RAD dan fase evaluasi, yang melibatkan analisis dan pengguna, terdiri dari empat fase dilihat pada gambar 2.3 sebagai berikut:



Gambar 2.3 Metode Rapid Application Development

1. Fase perencanaan

Fase ini merupakan fase diskusi antara penganalisis dan pengguna untuk mengidentifikasi tujuan dari sistem yang dibuat dan kebutuhan informasi yang muncul untuk mencapai tujuan tersebut, serta menganalisis sistem yang dibutuhkan pengguna. Fase ini bertujuan untuk memecahkan masalah yang ada

2. Fase membuat *prototype*

Langkah selanjutnya adalah membuat *prototype*. Pengembang akan membuat *prototipe* aplikasi yang diinginkan dengan fitur dan fungsi yang berbeda. Tujuannya adalah untuk memeriksa bahwa *prototype* yang diproduksi sesuai dengan tujuan yang diharapkan. Meski begitu,

pada tahap ini bisa dilakukan lebih dari satu kali. Terkadang juga melibatkan calon pengguna yang melakukan tes dan memberikan umpan balik. Proses ini memungkinkan peneliti untuk menyelidiki kemungkinan kesalahan di masa depan. Ini berguna untuk mengurangi kesalahan dan debugging. Langkah ini juga bekerja sedemikian rupa sehingga sistem akan dibuat lebih user-friendly, mudah digunakan, stabil dan tidak sering rusak saat digunakan nanti.

3. Fase proses pengembangan

Setelah prototype sistem informasi dibuat, langkah selanjutnya adalah mengubah prototype tersebut menjadi sebuah sistem informasi dengan menggunakan bahasa pemrograman. Oleh karena itu, langkah ini cukup intens. Peneliti terus melakukan coding aplikasi, melakukan pengujian sistem dengan menggunakan alat dan kerangka kerja yang mendukung RAD sehingga dapat diimplementasikan dengan cepat.

4. Fase implementasi aplikasi

Pada tahap ini peneliti memperbaiki kekurangan-kekurangan yang mungkin timbul pada saat pengembangan aplikasi, langkah terakhir ini dilakukan sebelum pengajuan aplikasi kepada pelanggan.

2.9 Algoritma Genetika

Algoritma genetika pertama kali dikembangkan oleh universitas John Holland 1975 Michigan. John Holland mengatakan tidak ada masalah Adaptasi (alami atau buatan) dapat dirumuskan istilah terminologi genetik. Algoritma genetika adalah proses simulasi evolusioner Rekayasa Genetika Darwin dan Kromosom (Demasya, 2018).

Teknologi pencarian Dilakukan oleh algoritma genetika dalam kombinasi dengan solusi yang mungkin disebut populasi. Orang-orang dalam kelompok disebut dari sudut kromosom. kromosom ini adalah solusi yang stagnan bentuk simbolis. Populasi awal dibuat secara acak. Hasil evolusi kromosom melalui proses yang disebut generasi. Pada

setiap generasi, kromosom melewati suatu proses evaluasi menggunakan alat ukur yang disebut fungsi fitness. Nilai dari berbagai kromosom menunjukkan kualitas kromosom dari populasi ini. Generasi selanjutnya disebut anak (*offspring*). Terbentuk dari dua kromosom generasi sekarang yang berfungsi sebagai kromosom dapat ditingkatkan menggunakan operator crossover memodifikasinya operator mutasi. Populasi generasi itu baru dibentuk dengan memilih fitness dari kromosom induk (parent) ukuran populasi (karena kesesuaian kromosom anak (keturunan) dan penolakan kromosom lain populasi) konstan. Setelah beberapa generasi, algoritma ini konvergen ke kromosom tertinggi.

2.9.1 Struktur Algoritma Genetika

Struktur umum algoritma genetika (Demasya, 2018) adalah sebagai berikut:

a) Pembangkitkan populasi awal

Generasi dari populasi awal Populasi awal ini dibangkitkan secara acak untuk memberikan solusi awal. Populasi itu sendiri terdiri dari serangkaian kromosom yang mewakili solusi yang diinginkan.

b) Pembentukan generasi baru

Pembentukan generasi baru menggunakan tiga operator yaitu operator replikasi/seleksi, crossover, dan mutasi. Proses ini diulang untuk mendapatkan kromosom yang cukup untuk membentuk generasi baru. Generasi baru ini merupakan representasi dari solusi baru.

c) Evaluasi solusi

Proses ini mengevaluasi di setiap populasi dengan cara menghitung nilai fitness untuk setiap kromosom dan menskor sampai kriteria penangkapan terpenuhi. Jika kriteria terminasi tidak terpenuhi, pengulangan langkah 2 akan membentuk generasi baru.

Beberapa kriteria berhenti yang sering digunakan sebagai berikut:

1. Berhenti pada generasi tertentu.
2. Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness tertinggi tidak berubah
3. Berhenti bila dalam n generasi berikut tidak didapatkan nilai fitness yang lebih tinggi.

2.9.2 Komponen Algoritma Genetika

Komponen utama dari algoritma genetika (Demasya, 2018) sebagai berikut:

a) Teknik Pengodean

Pengodean merupakan suatu metode penempatan populasi awal pada suatu kromosom sebagai kandidat solusi suatu masalah, yang merupakan kunci dari masalah tersebut bila menggunakan algoritma genetika. Teknik pengkodean ini termasuk pengkodean gen dan kromosom. Gen adalah bagian darinya Kromosom. Gen biasanya mewakili variabel.

b) Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis-jenis operator genetika yang akan diimplementasikan. Menurut ukuran populasi setelah ditentukan, kemudian kromosom perlu diinisialisasi terhadap populasi. Inisialisasi kromosom dilakukan secara acak. Namun, tetap perlu memperhatikan area solusi dan batasan masalah yang ada.

c) Evaluasi fitness

Merupakan dasar dari proses seleksi. Artinya, string diubah menjadi parameter fungsi, fungsi targetnya dievaluasi, dan kemudian ubah fungsi tujuan menjadi fungsi fitness. Di mana fungsi fitness digunakan sebagai

dasar untuk memaksimalkan masalah seleksi populasi generasi berikutnya. Untuk masalah minimasi nilai goodness-of-fit adalah kebalikan dari nilai minimum yang diharapkan.

Rumus fitness persamaan 2.1 yang digunakan adalah:

$$\text{Fitness} = \frac{1}{1 + \sum_{i=1}^n f_i}$$

d) Seleksi

Proses seleksi bertujuan untuk menyeleksi populasi yang terpilih untuk proses tersebut penyilangan dan mutasi untuk mendapatkan calon induk yang baik. Induk yang baik akan menghasilkan keturunan yang baik. Langkah pertama dari seleksi adalah pencarian nilai fitness. Setiap populasi wadah pemilihan menerima probabilitas replikasi dependen nilai objektifnya sendiri menjadi nilai objektif dari setiap individu. Nilai fitness ini akan digunakan nanti tahap seleksi selanjutnya.

e) Crossover

Crossover adalah proses persilangan dua kromosom menghasilkan kromosom anak yang diharapkan lebih baik (*offspring*) induk. Crossover bertujuan untuk membuat peningkatan variasi senar . Populasi crossover interstring diperoleh dari yang sebelumnya. Prinsip crossover adalah melakukan operasi (pertukaran, aritmatika). Mencocokkan gen dari dua gen untuk membuat individu baru.

f) Mutasi

Mutasi adalah proses perubahan nilai dari satu atau lebih gen Kromosom. Individu yang telah lulus prosedur seleksi dan crossover . Menghasilkan individu baru (keturunan) yang bermutasi . Ini berkontribusi untuk mempercepat munculnya perbedaan individu dalam populasi. Mutasi

ini memainkan peran dalam menggantikan gen yang hilang oleh ekskresi . Pilihan yang memungkinkan reproduksi gen yang tidak ditemukan di inisialisasi populasi.

2.10 Use Case

Use case adalah diagram yang memungkinkan kita untuk menggambarkan kemungkinan skenario penggunaan yang dikembangkan sistem. Ini mengungkapkan apa yang harus dilakukan sistem tetapi tidak membahas rincian realisasi (Seidl et al., 2015),.

Use Case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat, use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan yang berhak menggunakan fungsi-fungsi tersebut (Sukamto dan Shalahuddin., 2014).

Use Case diagram memiliki beberapa simbol dilihat pada tabel 2.1 sebagai berikut :

Tabel 2.1 Simbol Use Case

No	Simbol	Keterangan
1	Use Case 	Fasilitas yang disediakan oleh sistem sebagai entitas yang bertukar pesan dengan entitas atau aktor lain. Gunakan kata kerja di awal nama use case untuk mengekspresikannya.
2	Aktor ♀ 	Seseorang, proses, atau sistem lain di luar sistem informasi yang sedang dibuat yang berinteraksi dengan sistem yang dibuat.
3	Asosiasi 	Komunikasi antara aktor dan use case, atau antara use case dan aktor.
4	Generalisasi 	Hubungan generalisasi-spesialisasi antara dua kasus penggunaan di mana satu fitur lebih umum daripada yang lain (umum-spesifik).
5	Ekstensi / <i>Extend</i> 	Hubungan use case tambahan untuk use case. Kasus penggunaan tambahan dapat berdiri sendiri tanpa

		kasus penggunaan tambahan.
6	<p><i>Include</i></p> <p>-----></p> <p><i>Uses</i></p> <p>—————></p>	<p>Hubungan kasus penggunaan tambahan dengan kasus penggunaan jika kasus penggunaan yang ditambahkan memerlukan kasus penggunaan tersebut untuk menjalankan fungsinya atau sebagai kondisi untuk mengeksekusi kasus penggunaan tersebut.</p>

2.11 Penelitian Terkait

Penelitian ini mencari dan mengumpulkan berbagai penelitian terdahulu untuk di jadikan referensi sebelum melakukan penelitian ada beberapa studi Relevan yang menggunakan algoritma Squeezer namun fokus pada aspek dan subjek yang berbeda adapun perbedaan penelitian terkait ataupun penelitian terdahulu dengan penelitian ini adalah

1. Pengembangan Aplikasi Penjadwalan Kegiatan Pelatihan Teknologi Informasi Dan Komunikasi Dengan Algoritma Genetika (STUDI KASUS: BPRTIK).

Penelitian ini dilakukan oleh Restie Maya Puspita , Arini , dan Siti Ummi Masrurah : 2016. Tujuan dibuat untuk membantu admin BPRTIK dalam mengolah data dan mencari solusi terbaik untuk menentukan jadwal kegiatan agar tidak bersinggungan.

2. Perancangan Sistem Penjadwalan Seminar Proposal Dan Sidang Skripsi Dengan Metode Algoritma Genetika.

Penelitian ini dilakukan oleh Dwi Oktarina dan Alyauma Hajjah : 2019. Tujuan dibuat untuk mempercepat proses penjadwalan dan dengan adanya informasi yang didapatkan dari web mempermudah mahasiswa dan dosen untuk mengetahui jadwal yang telah di ajukan, dan daftar jadwal menguji bagi dosen. Dengan diterapkannya Algoritma Genetika pada sistem penjadwalan seminar proposal dan sidang skripsi, proses kesalahan dan keterlambatan dapat diminimalisir.

3. Rancang Bangun Aplikasi Penjadwalan Kuliah Pada Stkip Widya Yuwana.

Penelitian ini dilakukan oleh Ridho Pamungkas : 2016. Tujuan dibuat untuk membantu lembaga tersebut dalam melakukan penjadwalan kuliah yang masih sering mengalami permasalahan yaitu sering terjadinya konflik antara data jadwal matakuliah, dosen dan ruangan.

4. Sistem Penyiraman Otomatis Berdasarkan Suhu Ruang Dan Kelembapan Tanah Berbasis Fuzzy Logic.

Penelitian ini dilakukan oleh Rahmalia Syahputri , Alexsander Hendra Wijaya , Nurfiana , Dodi Yudo Setyawan : 2022. Tujuan dibuat untuk memanajemen pertanian modern berbasiskan IoT. Hasil uji coba menunjukkan bahwa sistem otomasi yang dikembangkan ini dapat memberikan respon eksekusi penyiraman dengan baik berdasarkan data dari sensor.

5. Sistem Optimasi Aplikasi Penjadwalan Kuliah Menggunakan Algoritma Genetik

Penelitian ini dilakukan oleh Syahrul Mauluddin, Iskandar Ikbal, dan Agus Nursikuwagus : 2018. Tujuan dibuat untuk penjadwalan kuliah dengan algoritma genetik dapat membuat jadwal dengan optimal yakni sampai tidak ada jadwal yang bentrok.