

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu :

Harim Adi Saputro.dkk, “Implementasi Algoritma Genetika Untuk Optimasi Penggunaan Lahan Pertanian”, 2018, Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 12. . ada pun hasil penelitian ini adalah sebagai berikut : Algoritma genetika banyak digunakan dalam masalah optimasi dan mempunyai kemampuan untuk menghasilkan solusi yang baik untuk masalah-masalah rumit. Pada penelitian ini digunakan metode algoritma genetika untuk mendapatkan proporsi luas yang optimal pada setiap tanaman yang akan ditanam, sehingga pemanfaatan luas lahan pertanian dapat menghasilkan keuntungan yang maksimal dengan mengalokasikan luas lahan dan modal yang dimiliki petani.[1]

Awang Andhyka, “Penerapan Algoritma Genetika Pada Permasalahan Matematika”, 2018, Universitas Nahdlatul Ulama Sidoarjo, Sidoarjo. Adapun hasil pembahasan dalam penelitian ini adalah Dalam bidang era saat ini seperti teknologi informasi banyak permasalahan yang didapati, sehingga memunculkan ide supaya hal tersebut dapat diselesaikan. Permasalahan tersebut juga berhubungan dengan kehidupan sehari - hari. Sebagai contoh dalam lingkungan pendidikan, yaitu

permasalahan dalam matematika, yang sampai hari ini mejadi momok pada anak sekolah. Pemahaman tentang matematika dan penyelesaiannya membutuhkan waktu proses yang lama. Dari hal tersebut, ada beberapa cara penyelesaian permasalahan pada matematika, salah satunya dengan menggunakan suatu metode yaitu algoritma genetika. Sebagai contoh apabila digunakan untuk mencari permasalahan liniear kuadrat, garis ataupun kuadarat memudahkan dalam mencari penyelesaian. Algoritma genetika sendiri didasarkan pada kehidupan gen manusia yang berkembang setiap saat, dalam arti kata bawah penyelesaian menggunakan metode algoritma genetika yaitu membutuhkan nilai awal yang disebut individu, dengan pengertian lain semakin banyak individu awal yang digunakan semakin banyak punya kesempatan untuk menyelesaikan permasalahan. Dalam teori algoritma genetika, dibatasi denganya adanya batasan-batasan seperti pada kehidupan sehari hari, dimana batasan tersebut dapat digunakan sebagai nilai atau data yang membatasi supaya individu awal tidak melebihi batas. Algoritma genetika disarankan pada kehidupan manusia yang berkembang biak dengan diselingi adanya gangguan pada kehidupan sehingga menjadi lebih baik, dimana gangguan tersebut disebut dalam algoritma dengan kata sederhana[2]

David Kristiad, “Genetic Algorithm for Lecturing Schedule Optimization (Case Study: University of Boyolali)” IJCCS (Indonesian Journal of Computing and Cybernetics Systems), Yogyakarta. Adapun hasil pembahasan pada penelitian ini adalah GA adalah metode komputasi yang terinspirasi oleh seleksi alam. Perhitungan terdiri dari beberapa operator yaitu Seleksi Turnamen, Crossover Seragam, Penggantian Induk Lemahdan dua operator mutasi (Interchanging

Mutation and Violated Directed Mutation (VDM)). Metode dua mutasi dibandingkan untuk menemukan operator mutasi yang lebih baik. Perpustakaan direncanakan memiliki kemampuan untuk menentukan batasan kustom (persyaratan penjadwalan yang tidak diakomodasi oleh perpustakaan) tanpa modifikasi program inti. Hasil pengujian menunjukkan bahwa VDM lebih menjanjikan untuk solusi optimal daripada Interchanging Mutation. Di kasus UBY, solusi optimal (fitness value=1) dicapai dalam 12 menit 41 detik dengan menambahkan 6 ruangan baru dan tidak aktif 2 kendala yaitu, kuliah dimulai pukul 14.00 kecuali semester 3 program studi hukum sains dengan kelas pagi dan peserta kuliah tidak boleh melebihi kapasitas kelas.[3]

Arief Kelik Nugroho, (2019) Penerapan Algoritma Genetika Untuk Menyelesaikan Permasalahan Penjadwalan Dosen Dengan Fuzzy” Penjadwalan merupakan suatu permasalahan yang sangat kompleks dan memerlukan banyak waktu dalam proses pembuatannya, karena terdapat banyak batasan yang harus dipenuhi. Oleh karena itu permasalahan penjadwalan sangat sulit untuk dikerjakan dengan menggunakan metode konvensional algoritma genetika sebagai suatu metode untuk menyelesaikan masalah optimasi penjadwalan proyek dengan pemerataan sumber daya biaya dan kendala berupa waktu lintasan kritis. Adapun hasil dari penelitian ini adalah jadwal proyek memiliki pembiayaan yang lebih merata dengan menggunakan algoritma genetika dan CPM dibandingkan tanpa menggunakan algoritma genetika [4].

Samaher.dkk, 2018, Penerapan Algoritma Genetika Untuk Memaksimalkan Laba Produksi Jilbab, Journal of Environmental Engineering & Sustainable

Technology JEEST Vol. 02 No. 01, July 2018, Membahas tentang Proses perhitungan optimasi laba secara manual dirasa tidak mampu untuk memberi hasil yang akurat dalam waktu yang cepat. Oleh karena itu dibutuhkan algoritma yang tepat untuk memperoleh hasil optimasi laba maksimum yang akurat dan efisien. Pada penelitian ini akan membahas mengenai proses optimasi laba maksimum menggunakan algoritma genetika. Algoritma genetika memiliki kemampuan dalam menyelesaikan berbagai masalah kompleks (Mahmudy, 2013). Diharapkan algoritma genetika ini mampu menyelesaikan masalah optimasi laba pada proses produksi jilbab dan memberikan kualitas solusi yang lebih tinggi dibandingkan penelitian sebelumnya.[5]

Berdasarkan penelitian dari Wiga Ayu Puspaningrum (2018) dengan judul “Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS” —Penjadwalan mata kuliah merupakan kegiatan yang sangat penting untuk proses belajar mengajar di Jurusan Sistem Informasi. Proses belajar mengajar dilaksanakan oleh seluruh mahasiswa dan dosen yang mengajar, sehingga jadwal mata kuliah yang disusun harus dapat memfasilitasi kepentingan dosen dan mahasiswanya. Apabila beberapa batasan yang ada dalam penjadwalan tidak diperhitungkan dengan baik, maka akan menyebabkan sulitnya untuk melakukan penjadwalan mata kuliah. Batasan-batasan tersebut, antara lain mata kuliah yang diselenggarakan, jumlah kelas yang tersedia, jumlah waktu yang ada, dan ketersediaan dosen yang mengajar. Masalah-masalah penjadwalan yang terjadi di Jurusan Sistem Informasi tersebut dapat diminimalkan dengan perhitungan

penjadwalan yang tepat dan mempertimbangkan seluruh aspek yang berkaitan dengan kegiatan belajar mengajar di Jurusan Sistem Informasi.[6]

II.2. Uraian Teoritis

II.2.1. Aplikasi

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. Sedangkan pengertian penjualan adalah suatu proses seseorang atau organisasi untuk meyakinkan *customer* membeli produk yang ditawarkan. Aplikasi *mobile* dapat diartikan sebagai sebuah produk dari sistem komputasi *mobile*, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan yang komputasi kemampuan dapat digunakan saat mereka sedang dipindahkan. Contohnya adalah *personal digital assistant* (PDA), *smartphone* dan ponsel. [7]

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. [8]

II.2.2. Algoritma Genetika

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup; yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau “siapa yang kuat, dia yang bertahan (survive)”. Dengan meniru teori evolusi ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-pemmasalahan dalam dunia nyata. Ada 4 kondisi yang sangat mempengaruhi proses evolusi, yaitu :

1. Kemampuan organisme untuk melakukan reproduksi,
2. Keberadaan populasi organisme yang bias melakukan reproduksi,
3. Keberagaman organisme dalam suatu populasi dan
4. Perbedaan kemampuan untuk survive.[9]

II.2.3. Sejarah Algoritma Ganetika

Algoritma genetika pertama kali ditemukan oleh Jhon Holland dari Universitas Michigan pada awal 1970-an di New York, Amerika Serikat. Jhon Holland bersama murid muridnya serta rekan kerjanya lalu menghasilkan buku yang berjudul “Adaption in Natural and Artificial Systems” pada tahun 1975, yang cara kerjanya berdasarkan pada seleksi dan genetika alam. Konsep yang dipergunakan dalam algoritma genetika adalah mengikuti apa yang dilakukan oleh alam. Algoritma genetik khususnya diterapkan sebagai simulasi komputer dimana

sebuah populasi representasi abstrak (kromosom) dari solusi-solusi calon (individual) pada sebuah masalah optimisasi akan berkembang menjadi solusisolusi yang lebih baik. Secara tradisional solusi-solusi tersebut dilambangkan dalam biner sebagai string '0' dan '1', walaupun dimungkinkan juga penggunaan penyandian (encoding) yang berbeda. Evolusi dimulai dari sebuah populasi individual acak yang lengkap dan terjadi dalam generasi generasi. Dalam tiap generasi kemampuan keseluruhan populasi dievaluasi, kemudian multiple individuals dipilih dari populasi sekarang (current) secara stochastic (berdasarkan kemampuan mereka) lalu dimodifikasi (dengan mutasi atau rekombinasi) menjadi bentuk populasi baru yang menjadi populasi sekarang (current) pada iterasi berikutnya dari algoritma [9]

II.2.4. Aplikasi Algoritma Genetika

Algoritma genetika sudah banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter atau solusi yang optimal. Hal ini membuat banyak orang mengira bahwa algoritma genetika hanya dapat digunakan untuk menyelesaikan masalah optimasi saja. Namun, pada kenyataannya algoritma genetika juga memiliki kemampuan untuk menyelesaikan masalah-masalah selain optimasi. Algoritma genetika banyak diaplikasikan untuk berbagai macam permasalahan, yaitu :

1. Optimasi Beberapa penggunaan algoritma genetika untuk optimasi antara lain untuk optimasi numerik dan optimasi kombinatorial seperti Traveling Salesmen Problem (TSP), Perancangan Integrated Circuit atau IC, Job Scheduling, dan Optimasi video dan suara.

2. Pemrograman Otomatis Algoritma genetika untuk pemrograman otomatis antara lain untuk melakukan proses evolusi terhadap program komputer dalam merancang struktur komputasional, seperti cellular automata dan sorting networks.
3. Machine Learning Algoritma genetika juga telah berhasil diaplikasikan untuk memprediksi struktur protein. Algoritma genetika juga berhasil diaplikasikan dalam perancangan neural networks (jaringan syaraf tiruan) untuk melakukan proses evolusi terhadap aturan aturan pada learning classifier system atau symbolic production system dan dapat digunakan untuk mengontrol robot.
4. Model Ekonomi Dalam bidang ekonomi, algoritma genetika digunakan untuk memodelkan proses proses inovasi dan pembangunan bidding strategies.
5. Model Sistem Imunisasi Contoh penggunaan algoritma genetika dalam bidang ini untuk memodelkan berbagai aspek pada sistem imunisasi alamiah, termasuk somatic mutation selama kehidupan individu dan menemukan keluarga dengan gen ganda (multi gen families) sepanjang waktu evolusi.
6. Model Ekologis Algoritma genetika juga dapat digunakan untuk memodelkan fenomena ekologis seperti host-parasite co evolutions, simbiosis dan aliran sumber di dalam ekologi. [9]

II.2.5 Hal-Hal Yang Harus Dilakukan Dalam Algoritma Genetika

Beberapa hal yang harus dilakukan dalam algoritma genetika adalah :

1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
2. Mendefinisikan nilai fitness, yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
3. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti random-walk [10]

II.2.6. *Android Studio*

Android merupakan sistem operasi *mobile*. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data data ponsel sekalipun, atau data sistem sendiri. *Android* merupakan sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari *Android*, yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*
3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time enviroment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil. [11]

II.2.7 Tools Pembangunan *Android*

Untuk membangun sebuah sistem operasi *Android* dapat menggunakan Mac, *Windows* PC, ataupun *Linux*. *Tools* yang dibutuhkan gratis dan dapat di *download* dari *web*. Berikut adalah beberapa *tools* yang digunakan untuk membangun aplikasi *android*.

1. JDK (*Java Development Kit*)

JDK (*Java Development Kit*) adalah Paket fungsi API untuk bahasa pemrograman Java, meliputi *Java Runtime Environment* (JRE) dan *Java Virtual Machine* (JVM). JDK adalah Perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Envirotment*). JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan java. Tanpa adanya JDK maka kode-kode java sudah di buat tidak akan bisa di jadikan aplikasi berbasis Java.

2. Android SDK

Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis *Android*. Di dalamnya, terdapat beberapa *tools* seperti *debugger*, *software libraries*, *emulator*, dokumentasi, *sample code* dan tutorial.

3. ADT (*Android Development Tools*)

Android Development Tools (ADT) adalah *plugin* untuk Eclipse yang didesain untuk pengembangan aplikasi *Android*. ADT memungkinkan Eclipse untuk

digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan *framework* API Android, *debug* aplikasi, dan pemaketan aplikasi Android. [11]

II.2.8. Perkembangan Android

Perkembangan sistem *Android* berkembang sesuai versi keluarnya adapun versi *Android* dimulai dari :

a. Android versi 1.1

Android memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai gadget pada tanggal 9 Maret 2009 silam. Android versi 1.1 merupakan Android awal yang dimana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar muka bagi pengguna (*user interface*) yang lebih baik, serta beberapa aplikasi yang lain. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.11. Android versi 1.1

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

b. Android versi 1.5 (Cupcake)

Pada bulan Mei 2009 Android kembali mengalami perubahan versi. Android versi 1.1 kemudian disempurnakan dengan Android versi 1.5 atau yang dikenal sebagai Android Cupcake. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.12. Android versi 1.5 (Cupcake)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

c. Android versi 1.6 (Donut)

Donut (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana Android Cupcake, yaitu pada bulan September 2009. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.13. Android Versi 1.6 (Donut)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

d. Android versi 2.0/2.1 (Eclair)

Masih ditahun yang sama, Android kembali merilis operating sistem versi terbarunya, yaitu Android versi 2.0/2.1 Eclair. Android Eclair diluncurkan oleh Google 3 bulan setelah peluncuran. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.14. Android Versi 2.0/2.1 (Eclair)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

e. Android versi 2.2 (Froyo: Frozen Yoghurt)

Butuh 5 bulan bagi Google untuk melakukan regenerasi dari Android Eclair versi sebelumnya ke versi Froyo Frozen Yoghurt. Pada tanggal 20 Mei 2010, Android versi 2.2 alias Android Froyo ini dirilis. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.15. Android Versi 2.2 (Froyo: Frozen Yoghurt)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

a. Android versi 2.3 (Gingerbread)

7 bulan kemudian Android kembali melakukan gebrakan dengan merilis kembali Android versi 2.3 atau yang dikenal sebagai Android Gingerbread. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.16. Android Versi 2.3 (Gingerbread)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

g. Android versi 3.0/3.1 (Honeycomb)

Pada bulan Mei 2011 Android versi 3.0/3.1 atau Android Honeycom dirilis. Android Honeycomb merupakan sebuah sistem operasi Android yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis Android. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.17. Android Versi 3.0/3.1 (Honeycomb)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

h. Android versi 4.0 (ICS: Ice Cream Sandwich)

Android ICS atau Ice Cream Sandwich juga dirilis pada tahun yang sama dengan Honeycomb, yaitu pada bulan Oktober 2011. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.18. Android Versi 4.0 (ICS: Ice Cream Sandwich)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

i. Android versi 4.1 (Jelly Bean)

Android Jelly Bean merupakan versi Android yang terbaru pada saat ini. Salah satu gadget yang menggunakan sistem operasi Jelly Bean adalah Google Nexus 7 yang diprakarsai oleh ASUS, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.19. Android Versi 4.1 (Jelly Bean)

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

j. Android versi 4.4 (Kit Kat)

Kehadiran android kitkat merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau android akan meluncurkan OS baru yang bernama Android Key Lime Pie namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS Android KitKat yang sebagian besar orang sudah familiar dengan itu. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.20. Android Versi 4.4 (Kit Kat)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

k. Android versi 5.0.2 (Lollipop)

Android Lollipop merupakan keberadaan OS Android yang memang saat ini sudah menjadi trend baru di industri smartphone, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari OS tersebut. Kehadiran android versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi Lollipop ini bias lebih baik dibandingkan versi-versi sebelumnya. (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.21. Android Versi 5.0.2 (Lollipop)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

1. Android versi 6.0 (Marshmallow)

Android 6.0 Marshmallow adalah versi dari sistem operasi mobile Android. Pertama kali diperkenalkan Mei 2015 di Google I / O di bawah kode nama Android M, secara resmi dirilis pada Oktober 2015. Android Marshmallow memperkenalkan model izin aplikasi didesain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi (Kusniyati & Pangondian Sitanggang, 2019)



Gambar II.22. Android Versi 6.0 (Marshmallow)
Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

m. Android 7.0 Nougat

Diperkenalkan pada tanggal 19 Oktober 2019 Android 7.0 Nougat berfokus pada peningkatan performa *user interface* sehingga lebih intuitif dan penggunaan aplikasi secara bersamaan lebih banyak pada fitur *multi window*. Selain peningkatan fitur tadi, Android Nougat juga menambahkan beberapa fitur lain seperti dukungan cahaya malam atau mode malam, *keyboard default* yang dapat mengirim animasi GIF langsung dan dukungan panggilan *multi-endpoint*.



Gambar II.23. Android Versi 7.0 Nougat

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

n. Android 8.0 Oreo

Nama Oreo dipilih Android untuk digunakan pada versi Android 8.0 yang diluncurkan pada bulan Agustus 2017. *User interface* pada Android Oreo lebih simpel agar memudahkan dalam mengakses aplikasi. Pembaruan pada versi Oreo membawa beberapa fitur seperti fitur *Autofill* yang memberikan kemudahan dalam mengisi formulir misal, dukungan gambar dalam gambar dan pengoptimalan *booting* agar lebih cepat.



Gambar II.24. Android Versi 8.0 Oreo

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

o. Android 9.0 Pie

Android 9.0 Pie merupakan versi Android terbaru yang dirilis pada bulan Agustus 2018. Fitur unggulan yang dimiliki oleh versi 9.0 Pie ini adalah kemampuan AI atau kecerdasan buatan. Selain itu fitur lainnya yang diusung seperti *Adaptive Brightness* yang akan menyesuaikan kecerahan layar secara otomatis dan dukungan pada ponsel *bezel less*.



Gambar II.25. Android Versi 9.0 Pie

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)

p. Android 10

Versi Android terbaru diberi nama Android 10 untuk memperingati bahwa Android telah mencapai 1 dekade secara komersial. Versi Android 10 lebih berfokus pada penyempurnaan mode malam atau gelap serta peningkatan fitur *sound amplifier* untuk mengatur kualitas audio.



Gambar II.26. Android 10

Sumber : .(Kusniyati & Pangondian Sitanggang, 2019)[12]

II.2.9. MySQL

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat database yang bersifat *open source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *windows* maupun *linux*. MySQL juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. MySQL juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *phpMyAdmin* dengan aplikasi *XAMPP*. [13]

II.2.10. UML (*Unified Modelling Language*)

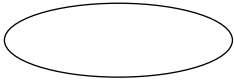
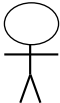
UML merupakan singkatan dari *Unified Modeling Language* yang merupakan sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi

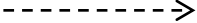
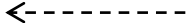
atau sistem yang tidak terdokumentasi harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya. [14]

1. Use Case Diagram

Use case diagram merupakan gambaran graphical dari beberapa atau semua aktor, *use case*, dan interaksi yang memperkenalkan suatu sistem. Simbol-simbol *use case diagram* dapat dilihat pada tabel II.1 *use case diagram*.

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan

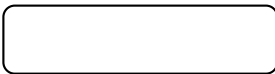
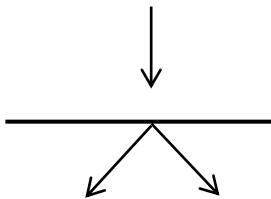
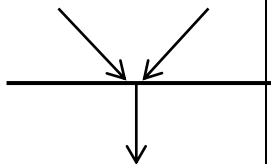
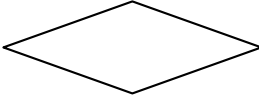
	<i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Janiver W. Janis; 2020)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram merupakan diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat. Simbol-simbol *activity diagram* dapat dilihat pada tabel II.2 *activity diagram*. Simbol- simbol yang digunakan dalam *activity diagram* yaitu:

Tabel II.2 Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

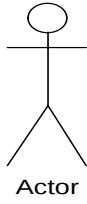
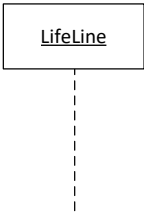
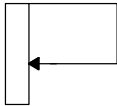
(Sumber : Janiver W. Janis; 2020)

3.Sequence Diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi objek yang berdasarkan urutan waktu. *Sequence diagram* dapat menggambarkan urutan atau

tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel II.3 *Sequence Diagram*

Gambar	Keterangan
	Proses atau sistem lain yang berinteraksi dengan sistem informasi.
	Merepresentasikan entitas tunggal dalam <i>sequence diagram</i> , digambarkan dengan kotak. Entitas ini memiliki nama, <i>Stereotype</i> atau berupa <i>instance</i> .
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Janiver W. Janis; 2020)

4. Class Diagram (Diagram Kelas)

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut *atribut* dan metode atau operasi. *Atribut* merupakan *variabel-variabel* yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

II.2.9. Normalisasi

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redudansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. [15]

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan grup. Adapun contoh bentuk tidak normal (Unnormal) dapat dilihat pada Tabel II.4 :

Tabel II.4. Contoh bentuk tidak normal (Unnormal)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2019

- 2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data. Adapun Contoh Bentuk Normal Pertama (1NF) dapat dilihat pada Tabel II.5 :

Tabel II.5. Contoh Bentuk Normal Pertama (1NF)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2019

- 3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key

sudah tergantung penuh terhadap key-nya. Adapun Contoh Bentuk Normal Pertama (1NF) dapat dilihat pada Tabel II.6.

Tabel II.6. Contoh Bentuk Normal Kedua (2NF)

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Sumber : Mukhlisulfatih Latief : 2019

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap *key*-nya. Adapun Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF) dapat dilihat pada Tabel II.7.

Tabel II.7 Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

Sumber : Mukhlisulfatih Latief : 2019