

BAB II LANDASAN TEORI

2.1. Tinjauan Studi

Dalam melaksanakan penelitian ini di ambil beberapa referensi dari jurnal terkait sebagai dasar penelitian aplikasi pemetaan SPBU di Kabupaten Jepara, di antaranya :

Penelitian pertama M Rukmana Yuda dan Ramdani Fatwa dalam jurnal pengembangan teknologi dan ilmu komputer yang berjudul Implementasi Algoritme *Dijkstra* pada *Webgis* untuk pencarian lokasi SPBU di kota Malang. Penelitian ini menggunakan aplikasi sistem informasi Geografis (SIG) berbasis web yang mampu menyalurkan informasi tentang jalur yang bisa ditempuh untuk mempercepat proses pencarian dengan mengikuti jalur rute terpendek menggunakan Algoritma Dijkstra untuk menghitung rute terpendek menggunakan Algoritma Dijkstra yang di hasilkan. Metodologi yang di lakukan di dalam penelitian ini yaitu studi literatur, penggumpulan data, penerapan *Shorttest Bestpath*, pembangunan *WebGis*, dengan menggunakan Algoritma Dijkstra untuk menghitung rute terpendek, pengujian dan analisis aplikasi yang sudah di bangun, dan menyimpulkan hasil penelitian. Aplikasi ini di terapkan *Bastpath* pada Algoritma Dijkstra dengan melakukan digitasi terhadap jalur provinsi atau jalur nasional di *overlay* dengan *googlemap* dengan tipe peta *roadmap* hingga mendapatkan akurasi yang akurat.[1]

Penelitian kedua Dwi, Ratnasari.dkk dalam jurnal ilmiah bidang teknologi informasi dan komunikasi yang berjudul Sistem Informasi Pencarian Tempat Kos Berbasis Android. Aplikasi ini merupakan sistem informasi bagi pendatang dari luar kota Mataram yang merasa kebingungan dalam mencari rumah kos yang sesuai, serta membantu pemilik kos dalam mempromosikan rumah kosnya secara *online* lewat *smartphone* berbasis android. Perancangan aplikasi ini di lakukan dengan menggunakan

metodologi waterfall, yaitu sebuah metode pengembangan sistem yang diawali dengan tahap analisis, kemudian dilakukan perancangan sistem dan implementasi pembuatan aplikasi serta mengujinya kepada pengguna. Sistem ini dikembangkan dengan menggunakan bahasa pemrograman Java serta XAMPP. Sistem ini menyajikan informasi detail tentang ketersediaan rumah kos yang sesuai dengan kebutuhan yang diinginkan, sehingga dapat menyakinkan pencari rumah kos sebelum memesan kamar rumah kos. Pesan notifikasi langsung dari sistem berupa informasi pemesanan kamar rumah kos dapat diterima oleh pemilik rumah kos dengan cepat. Sistem ini dibuat secara online dan interaktif, artinya sistem promosi ini membutuhkan bantuan dari pemilik rumah kos yang bertugas *me-maintenance* rumah kos yang mereka promosikan setiap saat. Terutama dalam hal *update* status di setiap kamarnya. Oleh karena itu sistem ini juga dapat digunakan langsung untuk update status tersebut. Pencari kos dapat mengetahui letak posisi rumah kos dan jarak pencari kos dengan rumah kos. [2]

Penelitian A, Herli Maximillian.dkk dalam *Journal of Information System Engineering and Business Intelligence* yang berjudul Sistem Pencarian Hotel Berdasarkan Rute Perjalanan Terpendek Dengan Mempertimbangkan Daya Tarik Wisata Menggunakan Algoritma Greedy. Pada penelitian ini dibangun sistem untuk melakukan pencarian hotel berdasarkan rute perjalanan terpendek dengan daya tarik wisata menggunakan Algoritma Greedy untuk memudahkan wisatawan dalam melakukan efisiensi jarak perjalanan wisata serta membantu dalam pemilihan hotel. Penelitian ini dilakukan melalui empat tahap, tahap pertama adalah pengumpulan data dan informasi daya tarik wisata dan hotel. Tahap kedua adalah analisa data dengan Algoritma Greedy serta melakukan penyesuaian penggunaan Algoritma berdasarkan karakteristik perjalanan yang dilakukan wisatawan. Tahap ketiga adalah pembangunan sistem, Tahap keempat adalah melakukan evaluasi sistem bersama para ahli yang telah berpengalaman dalam bidang pariwisata dan calon pengguna aplikasi ini. Sistem memberikan rekomendasi pilihan hotel dan urutan rute perjalanan wisata secara benar sesuai dengan pertimbangan jarak perjalanan. [3]

2.2. Tinjauan Pustaka

2.2.1 Pengertian Algoritma

Metode efektif yang diekspresikan sebagai rangkaian terbatas atau merupakan kumpulan perintah untuk menyelesaikan suatu masalah disebut algoritma. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan syarat untuk setiap permasalahan memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan sebuah algoritma. Algoritma juga memiliki pengulangan proses (iterasi), dan juga memiliki keputusan hingga keputusan selesai .[4]

2.2.2 Jalur Terpendek (*Shortest Path*)

Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (weighted graph), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar Kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya .[5]

Asumsi yang digunakan di sini adalah bahwa semua bobot bernilai positif. Lintasan terpendek adalah jalur yang dilalui dari suatu node ke node lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari node awal ke node akhir paling kecil. Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat lain. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot.[6]

Jalur terpendek adalah jalur minimum yang diperlukan untuk mencapai lokasi dari lokasi tertentu. Ada beberapa jenis masalah jalur minimum, yaitu:

- 1) Jalur minimum antara dua titik.
- 2) Jalur minimum antara semua pasangan titiknya.
- 3) Jalur minimum dari titik tertentu ke semua titik lainnya.

4) Jalur minimum antara dua titik yang melewati titik-titik tertentu.

Lintasan minimum tersebut dapat dicari menggunakan grafik. Grafik yang digunakan adalah grafik berbobot, yaitu grafik di mana setiap sisi menerima nilai atau bobot. Dalam hal ini, berat yang diharapkan adalah dalam bentuk jarak dan waktu .[7]

2.2.3 Optimasi

Optimisasi merupakan kumpulan prinsip-prinsip matematika dan metode yang digunakan untuk menentukan penyelesaian dari persoalan kuantitatif dalam banyak disiplin ilmu, termasuk fisika, biologi, teknik, ekonomi, dan bisnis. Banyak persoalan dapat dirumuskan dan dipecahkan dengan menggunakan sekumpulan gagasan dan metode terpadu yang membentuk bidang optimisasi.

Optimasi sering dimulai dengan serangkaian variabel dan parameter independen, mencakup persyaratan atau batasan pada nilai variabel independen yang dapat diterima. Batasan ini disebut sebagai kendala masalah. Elemen penting lain dari masalah optimisasi adalah ukuran "baik" yang disebut fungsi objektif yang tergantung pada variabel masalah. Solusi dari masalah optimisasi adalah himpunan nilai variabel yang memenuhi kendala, sehingga fungsi tujuan mencapai nilai optimal .[8]

2.2.4 Mobile SIG

2.2.1.1. Pengertian SIG (sistem informasi geografis)

SIG adalah sebuah integrasi cara kerja *software* maupun *hardware* untuk mengakses data dan layanan geospasial melalui *mobile* via jaringan kabel maupun nirkabel. GIS merupakan gabungan dari tiga teknologi, yaitu software SIG, teknologi GPS (Global Positioning System), dan perangkat telepon seluler yang membuat database dapat diakses oleh seseorang secara langsung atau real-time sehingga memudahkan pengguna mengetahui informasi lewat telepon seluler.[1]

2.2.1.2. Definisi SPBU

SPBU (Stasiun Pengisian Bahan Bakar untuk Umum) merupakan prasarana umum yang disediakan oleh PT. Pertamina untuk masyarakat luas guna memenuhi kebutuhan bahan bakar. [1] Pada umumnya SPBU menjual bahan bakar sejenis premium, solar, pertamax dan pertamax plus. Untuk SPBU sendiri dibagi menjadi 3 jenis usaha yaitu :

a) **COCO (Company Operation Company Owner)**

Merupakan SPBU yang di miliki dan di kelola oleh Pertamina. Dalam hal ini yang mengelola adalah PT. Pertamina Retail sebagai anak perusahaan. Saat ini sudah banyak tersebar SPBU coco di Indonesia.

b) **DODO (Dealer Operation Dealer Owner)**

Merupakan SPBU murni milik swasta atau perorangan. Jadi segala hal mengenai manajemen perusahaan di kelola oleh perorangan atau badan usaha.

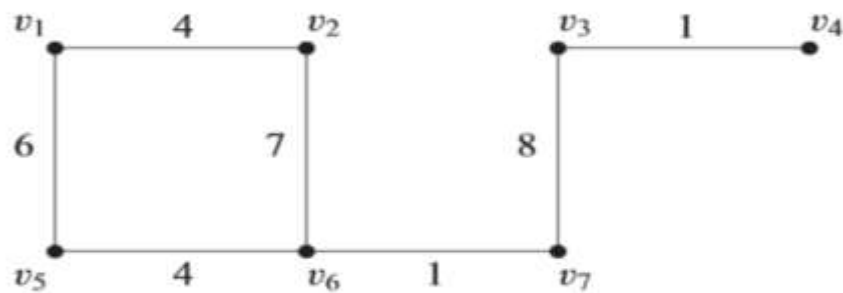
c) **CODO (Company Operation Dealer Owner)**

Merupakan SPBU milik swasta atau perorangan yang bekerjasama dengan PT Pertamina Retail.

2.2.5 Algoritma Greedy

Algoritma Greedy merupakan algoritma yang membentuk solusi langkah perlangkah dengan mencari nilai maksimum sementara pada setiap langkahnya. Dalam menyelesaikan permasalahan, Algoritma Greedy melakukannya secara bertahap.[9] Tahap penyelesaiannya adalah:

1. Mengambil pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan.
2. Berharap bahwa dengan memilih optimum local pada setiap langkah akan berakhir dengan optimum global.



Gambar 2.1. Diagram Alur Algoritma Greedy [9]

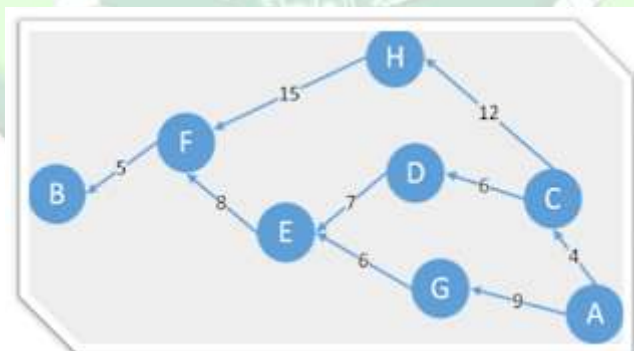
Algoritma Greedy memiliki pendekatan untuk membangun solusi secara bertahap melalui urutan yang terus berkembang sampai solusi dari masalah telah tercapai.[10] Persoalan optimasi dalam konteks Algoritma Greedy disusun oleh komponen-komponen sebagai berikut:

- a) Himpunan Kandidat (C) : Merupakan himpunan yang berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.
- b) Himpunan Kandidat, (C) : Himpunan Kandidat merupakan himpunan yang berisi mengenai elemen pembentuk Solusi.
- c) Himpunan Solusi, (S):Himpunan Solusi merupakan himpunan-himpunan yang berisi elemen solusi pemecahan masalah.
- d) Fungsi Seleksi: Fungsi Seleksi adalah fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
- e) Fungsi Kelayakan: Fungsi Kelayakan merupakan fungsi yang memeriksa apakah suatu kandidat yang dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.
- f) Fungsi Obyektif :Fungsi Obyektif merupakan fungsi yang memaksimalkan atau meminimumkan nilai solusi. [3]

Untuk mencari rute tercepat atau lintasan terpendek dalam Algoritma Greedy dilakukan dengan rumus berikut :

1. Pertama lakukan periksa sisi yang berhubungan dengan sisi pertama (lokasi pertama). Pilih sisi yang memiliki bobot (jarak) terkecil dengan sisi berikutnya. Jarak tersebut merupakan menjadi rute terpendek pertama, dapat dimisalkan sebagai (L1).
2. Selanjutnya menentukan rute terpendek berikutnya dapat dimisalkan (D) dengan cara sebagai berikut :
 - a) Untuk mencari rute berikutnya dengan rumus $D(i) = L1 + \text{bobot sisi berikutnya}$. Dijelaskan jarak pertama dijumlahkan dengan bobot berikutnya. Jika terdapat rute lainnya, maka lakukan penjumlahan jarak rute tersebut dengan jarak sebelumnya.
 - b) Pilih $D(i)$ yang memiliki jarak terkecil, kemudian jika terdapat rute lainnya lakukan perbandingan. Jika rute lainnya tersebut memiliki jarak lebih kecil daripada $D(i)$, maka rute tersebut merupakan rute yang akan diperhitungkan selanjutnya.
 - c) Untuk mencari lintasan terpendek berikutnya dapat dilakukan dengan menggunakan cara kedua di atas.

Contoh Algoritma Greedy mencari jarak terpendek dari peta



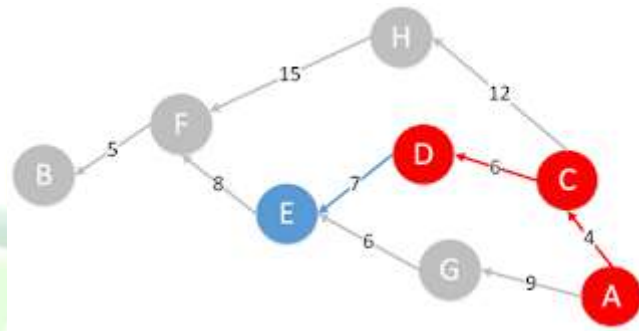
Gambar 2.2 Contoh Algoritma Greedy

Misalkan kita ingin bergerak dari titik A ke titik B, dan kita telah menemukan beberapa jalur dari peta.

Untuk mencari jarak terpendek dari A ke B, sebuah algoritma greedy akan menjalankan langkah-langkah seperti berikut:

Kunjungi satu titik pada graph, dan ambil seluruh titik yang dapat dikunjungi dari titik sekarang.

1. Cari local maximum ke titik selanjutnya.
2. Tandai graph sekarang sebagai graph yang telah
3. dikunjungi, dan pindah ke local maximum yang telah ditentukan.
4. Kembali ke langkah 1 sampai titik tujuan didapatkan.



Gambar 2.3. *Graph* Algoritma Greedy

Dengan menggunakan algoritma greedy pada graph di atas hasil akhir jarak terpendek adalah ACDEFB. Hasil jarak terpendek ini sebenarnya tidak tepat dengan jarak pendek sebenarnya (A-G-E-F-B). Maka dari algoritma yang tidak selamanya benar namun algoritma yang mendekati nilai kebenaran.

Pemecahan Masalah dengan Algoritma Greedy

Strategi greedy untuk memilih job:

Pada setiap langkah, pilih job i dengan p_i yang terbesar untuk menaikkan nilai fungsi obyektif F .

Contoh:

$$(p_1, p_2, p_3, p_4) = (50, 10, 15, 30)$$

$$(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$$

Tabel 2.1. Kelayakan *Algoritma Greedy*

Langkah	J	$F = \sum p_i$	Keterangan
0	$\{\}$	0	-
1	$\{1\}$	50	layak
2	$\{4,1\}$	$50 + 30 = 80$	layak
3	$\{4, 1, 3\}$	-	tidak layak
4	$\{4, 1, 2\}$	-	tidak layak

Solusi optimal: $J = \{4, 1\}$ dengan $F = 80$

```

function JobScheduling1(input C : himpunan_job) → himpunan_job
f Menghasilkan barisan job yang akan diproses oleh mesin }

Deklarasi
i : integer
J : himpunan_job   f solusi }

Algoritma
J ← {}
while C ≠ {} do
  i ← job yang mempunyai p[i] terbesar
  C ← C - {i}
  if (semua job di dalam J ∪ {i} layak) then
    J ← J ∪ {i}
  endif
endwhile
f C = {} }
return J

```

Gambar 2.4. Kode Algoritma *Greedy*

2.2.6 Algoritma Dijkstra

Algoritma yang digunakan untuk menentukan jarak terpendek dari satu vertex ke vertex yang lainnya pada suatu graph berbobot, jarak antar vertex adalah nilai bobot dari setiap edge pada graph. Suatu bobot harus bernilai positif (bobot ≥ 0). Algoritma Dijkstra ditemukan oleh Edger Wybe Dijkstra. Algoritma Dijkstra dikenal juga sebagai algoritma greedy yaitu algoritma yang penyelesaian masalah dengan mencari nilai maksimum. Cara kerja algoritma Dijkstra dalam pencarian jarak terpendek adalah perhitungan dari vertex asal ke vertex terdekatnya, kemudian ke vertex yang kedua, dan seterusnya .[11]

Menurut Fakhri (2008), di dalam melakukan perhitungan dengan algoritma Dijkstra ada beberapa skema umum yang digunakan pada pencarian jarak terpendek, antara lain:

1. Membuat 3 buah list, yaitu list jarak (list 1), list simpul-simpul sebelumnya (list 2) dan list yang sudah dikunjungi (list 3), serta sebuah variabel untuk menampung simpul pada saat ini (current vertex).
2. Dalam list jarak, di isi dengan nilai tak hingga, kecuali simpul awal yang diisi dengan nilai 0.
3. Pada list 2, diisi dengan false.
4. Pada list 3, diisi dengan null.
5. Current vertex diisi dengan simpul awal (start).
6. Menandai current vertex sebagai simpul yang telah dikunjungi.
7. Update list 1 dan 2 berdasarkan simpul-simpul yang dapat langsung dicapai dari current vertex.
8. Update current vertex dengan simpul yang paling dekat dengan simpul awal.
9. Lakukan langkah no 6 sampai semua simpul telah dikunjungi.

Algoritma Dijkstra memiliki kompleksitas $O(n^2)$. Sehingga total waktu asimptotik komputasi dalam mencari pasangan vertex terpendek adalah $T(n) = n \cdot (n^2) = O(n^3)$. Dengan demikian algoritma Dijkstra memiliki keuntungan lebih dalam hal running time. [12]

2.2.7 Google Maps

Merupakan layanan dari *Google* yang mempermudah penggunaanya untuk melakukan kemampuan pemetaan untuk aplikasi yang di buat. Sedangkan *Google Maps API* (Application Progaming Interface) adalah fungsi-fungsi pemrograman yang disediakan oleh layanan atau aplikasi agar bisa diintegrasikan ke dalam Web atau aplikasi mobile yang sedang di buat. Google Maps Api merupakan sebuah fitur yang disediakan oleh Google Maps untuk memudahkan pengguna menyimpan (embed) Google Maps di aplikasi dan menentukan titik poin tertentu pada peta yang dimiliki Google Maps. [2]

2.2.8 API Google Maps

API Google Maps adalah antarmuka aplikasi yang dapat diakses melalui JavaScript sehingga Google Maps dapat ditampilkan pada halaman web yang sedang dibuat. Menggunakan Google Maps API menghemat waktu dan uang untuk membuat aplikasi peta digital yang andal, karena pengguna hanya fokus pada data yang ingin mereka sesuaikan, karena data peta sudah disediakan oleh Google Maps. [13] Ada 2 cara untuk mengakses data dari Google Maps, tergantung pada data yang ingin kita pulihkan dan dekripsi dari Google Maps, adalah sebagai berikut:

1. Tidak harus menggunakan API Key untuk akses data dari Google Maps.
2. Tidak harus menggunakan API key untuk akses data dari Google Maps.
3. Untuk pendaftaran API key harus dilakukan dengan data registrasi dalam bentuk nama domain web yang kita buat.

2.2.9 GPS

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang terdiri dari setidaknya 24 satelit. GPS berfungsi dalam segala kondisi cuaca, di mana pun di dunia, 24 jam sehari, tanpa biaya berlangganan atau biaya penyiapan. Penelitian ini dimulai dari pengumpulan data dan pemetaan lokasi SPBU di Kabupaten Jepara. Data lain yang dibutuhkan adalah data jalan, persimpangan, dan info jalur satu arah. Data tersebut akan digunakan untuk penggambaran peta lokasi SPBU. Pencarian data yang dibutuhkan sebagian besar adalah data spasial, sehingga dibutuhkan GPS, metode tracking, dan wawancara kepada pemilik/karyawan SPBU. Perhitungan jarak dilakukan oleh sistem dengan memilih posisi user pada peta. Perhitungan jarak ini akan menghasilkan rekomendasi SPBU terdekat dari posisi user. [14]

2.2.10 Android Studio

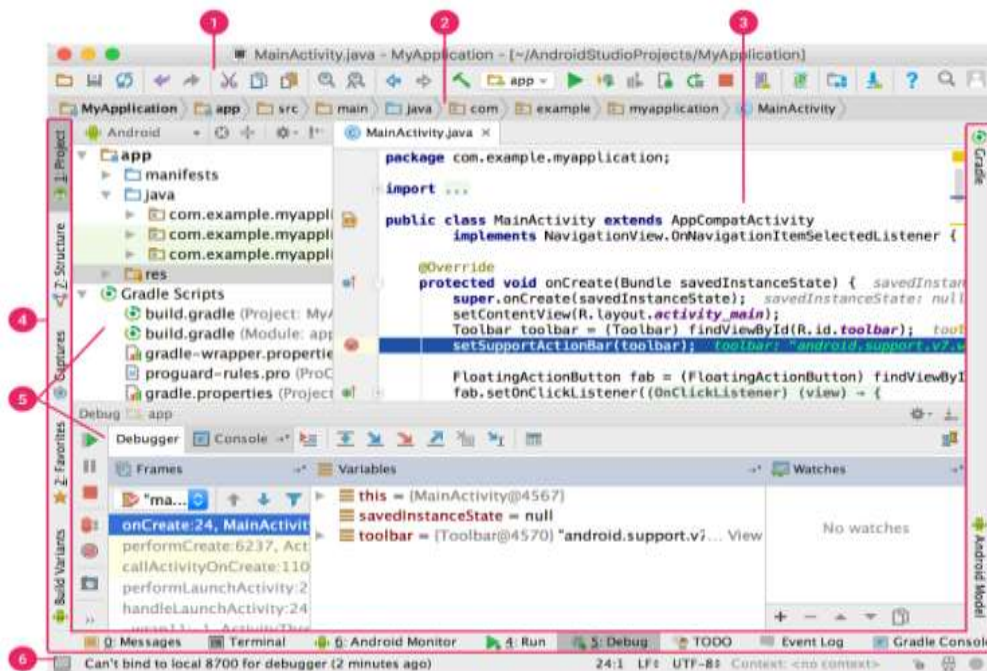
Android Studio merupakan IDE (Integrated Development Environment) resmi untuk platform Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada konferensi *Google I/O* oleh Produk Manager Google, Elite Power. Android Studio bersifat free dibawah Apache Licence

2.0. Android Studio diawali dengan versi 0.1 pada bulan Mei 2013, kemudian dibautkan versi 0.8 pada bulan Juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, lalu dimulai dari versi 1.0 Berbasiskan *JetBrain's IntelliJ IDE* dan di desain khusus untuk Android Development.

Android Studio sendiri dikembangkan berdasarkan *IntelliJ IDEA* yang mirip *Eclipse* disertai dengan ADT plugin (Android Development Tools). Android Studio memiliki fitur sebagai berikut :

- A. Projek berbasis pada Gradle Build
- B. Refactory dan pembenahan bug yang cepat
- C. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- D. Mendukung Proguard And App-signing untuk keamanan.
- E. Memiliki GUI aplikasi android lebih mudah.
- F. Didukung oleh Google Cloud Platform untuk setiap aplikasi yang dikembangkan.

Namun dari segi grafik Android Studio lebih unggul dibandingkan Eclipse, tetapi Android Studio jauh lebih berat dibandingkan dengan Eclipse karena membutuhkan spesifikasi komputer yang cukup tinggi, dan Android Studio dikhususkan untuk pengembangan Android Studio saja dibandingkan Eclipse (tidak multi programming), jadi Android Studio lebih nyaman dalam pengkodingan untuk membuat atau mengembangkan sebuah aplikasi



Gambar 2.5 Jendela utama *Android Studio*

Jendela utama pada Android Studio terdiri dari beberapa bidang logika yang diidentifikasi pada gambar :

1. Bilah Alat : memungkinkan developer untuk melakukan berbagai jenis tindakan, termasuk menjalankan aplikasi dan meluncurkan program Android.
2. Bilah Navigasi : membantu developer bernavigasi antara proyek dan membuka file untuk di edit. Bilah ini memberikan tampilan struktur yang ringkas dalam jendela Project.
3. Jendela Editor : tempat developer membuat dan memodifikasi kode bergantung pada jenis file, editor dapat berubah.. Seperti saat melihat file tata letak, editor menampilkan Layout Editor.
4. Bilah Jendela Alat : muncul saat di luar jendela IDE dan berisi tombol yang memungkinkan developer meluaskan atau mengecilkan jendela alat.
5. Jendela Alat : memberi developer akses ke tugas tertentu seperti pengelolaan proyek, pencarian atau penelusuran, kontrol versi, dan sebagainya.
6. Bilah Status : menampilkan status proyek dan IDE itu sendiri serta setiap peringatan.[15]

2.2.11 Android

Android merupakan sistem operasi berbasis Linux untuk telepon pintar terpopuler yang dikeluarkan oleh Google. Android juga memiliki fitur dan tampilan baik serta digunakan alat multimedia dan dapat mengoperasikan perangkat keras seperti sensor dan lain sebagainya. Sejak rilis tahun 2008, hingga 2018 Android sudah merilis beberapa versi dengan versi-versi terbaru yaitu Android Pie. Pada tahun 2016 Android sudah memiliki lebih dari 2 miliar pengguna aktif tiap bulannya. Lisensi kode sumber Android memiliki lisensi open source. Sehingga, dapat menarik para komunitas pengembangan untuk menggunakan Android dalam projek komunitasnya. [15]

Tabel 2.2 Versi Android

No	Nama Kode	No. Versi	Tanggal Rilis	API Level
1	Alpha	-	Awal Tahun 2007	-
2	Beta	-	November 2007	-
3	Angel cake	1.0	23 September 2008	1
4	Bettenberg	1.1	9 Februari 2009	2
5	Cupcake	1.5	27 April 2009	3
6	Donut	1.6	15 September 2009	4
7	Éclair	2.0 – 2.1	26 Oktober 2009	5-7
8	Froyo	2.2 – 2.3	20 Mei 2010	8
9	Gingerbread	2.3 – 2.3.7	6 Desember 2010	9-10
10	Honeycomb	3.0 – 3.2.6	22 Februari 2011	11-13
11	Ice Cream Sandwich	4.0 – 4.0.1	18 Oktober 2011	14-15
12	Jelly Bean	4.1 – 4.3.1	9 Juli 2012	16-18
13	Kitkat	4.4 – 4.4.4	31 Oktober 2013	19-20
14	Lollipop	5.0	15 Oktober 2014	21-22
15	Marshmallow	6.0	5 Oktober 2015	23
16	Nougat	7.0	22 Agustus 2016	24
17	Nougat	7.1	4 Oktober 2016	25
18	Oreo	8.0	21 Maret 2017	26

2.2.12 Kotlin

Kotlin merupakan Bahasa pemrograman dengan pengetikan statis dan dapat berjalan pada mesin Virtual Java, atau menggunakan Compiler LLVM yang dapat menyusun ke dalam bentuk kode sumber JavaScript. Memiliki keunggulan untuk mudah dipelajari, sederhana, dan efisien. Kotlin diumumkan sebagai salah satu Bahasa pemrograman yang dapat digunakan untuk membuat aplikasi Android pada Google I/O 2017 berdampingan dengan Java dan C++ yang telah terlebih dahulu sebagai Bahasa resmi untuk pengembangan aplikasi Android. [16]

Berikut adalah beberapa perbaikan dari Java yang ada di Kotlin :

- A. Null reference kini dikontrol oleh type sistem
- B. Tidak ada raw type
- C. Array di Kotlin tidak berubah
- D. Kotlin memiliki function type yang layak
- E. Use-site variance tanpa perlu wildcard
- F. Kotlin tidak memiliki checked exception

2.2.13 Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial.[17] Terdapat dua bentuk pertanyaan dalam skala likert, yaitu bentuk pertanyaan positif untuk mengukur skala positif, dan bentuk pertanyaan negatif untuk mengukur skala negatif. Pertanyaan positif diberi skor 5, 4, 3, 2, dan 1; sedangkan bentuk pertanyaan negatif diberi skor 1, 2, 3, 4, dan 5.[18]

2.2.14 Firebase

Firebase merupakan BaaS (Backend as a Service) suatu layanan dari google untuk memudahkan para pengembang aplikasi untuk mengembangkan aplikasinya. Dengan adanya firebase app, developer bisa fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend.

Pada tahun 2014, Google mengaku sisi perusahaan yang berbasis di San Fransisco bernama Firebase, Inc. Dengan menyediakan berbagai solusi pengembangan yang dirancang untuk integrasi fitur berbasis cloud ke dalam aplikasi sululer dan web. Setelah memberi perusahaan, Google menggabungkan layanan yang disediakan oleh Firebase dengan sejumlah fitur pelengkap yang sebelumnya termasuk dalam bagian dalam dari Google Cloud Platform. Fitur yang sebelumnya tergabung dari dua platform adalah apa yang sekarang dikenal sebagai Firebase. [16]

Adapun fitur yang dimiliki firebase, diantaranya :

1. *GoogleAnalytics*

Analityc ini memberikan data seputar perilaku pengguna pada aplikasi Android dan iOS agar pengguna dapat mengambil keputusan yang lebih baik tentang informasi produk dan pengoptimaalan pemasaran.

2. *Real-Time Database*

Untuk menyimpan dan sinkronkan data antara pengguna dan perangkat secara *realtime* menggunakan *database* noSQL yang di *hosting* secara *cloud*.

3. *Authentication*

Untuk mengelola pengguna dengan cara yang mudah dan aman.

4. *Cloud Storage*

Untuk menyimpan dan membagikan gambar, audio, video, atau konten lain secara mudah dengan menyimpan objek yang handal, sederhana, dan hemat biaya.

5. *Hosting*

Memudahkan *hosting* *Web* statis dengan fitur yang dibuat khusus untuk *Web* aplikasi modern.

Ada pula struktur *Firestore* yang berbeda dengan MySQL diantaranya :

- A. *Firestore Realtime Database* bertipe NoSQL, sedangkan MySQL bertipe RDBMS.
- B. *Firestore Realtime Database* bersifat Realtime, sedangkan MySQL tidak Realtime.
- C. Dengan *Firestore Realtime Database* tidak perlu membuat koding di sisi server.
- D. Koding yang dipakai *Firestore* untuk mengolah data kompleks yang lebih banyak dan rumit daripada koding yang dipakai untuk mengolah data MySQL.
- E. *Firestore Realtime Database* punya versi berbayar, sedangkan MySQL tidak berbayar atau gratis sepenuhnya.

2.2.15 Unified Modeling Language

UML (Unified Modelling Language) merupakan visualisasi dan dokumentasi hasil analisa dan desain sintak dalam memodelkan sistem secara visual, serta merupakan alat perancangan sistem yang berorientasi pada objek. UML (Unified Modeling Language) ditujukan untuk membantu tim pengembang berkomunikasi, eksplorasi potensi desain, dan validasi desain arsitektur perangkat lunak. UML (Unified Modeling Language) memiliki tiga kategori utama yaitu behaviour diagram, struktur diagram, dan interaction diagram, yang dimana pada setiap kategori diagram menjelaskan arsitektur sistem dan saling terintegrasi. [15]

2.2.16 Use Case Diagram

Use Case Diagram merupakan deskripsi fungsi dari sistem berdasarkan perspektif pengguna, dan merupakan sebuah dokumentasi yang digunakan sebagai proses analisa untuk menangkap permintaan sistem dan memahami sistem bekerja. Use Case Diagram menetapkan apa yang dikerjakan oleh sistem seperti kebutuhan fungsional dan tidak untuk menentukan kebutuhan non-fungsional. [19]

Berikut adalah bentuk simbol dan fungsi dari sistem use case tersebut :

Tabel 2.3 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		Use Case	Deskripsi dari urutan aksi yang ditampilkan sistem.
2.		Collaboration	Interaksi elemen lain yang bekerja sama untuk menyediakan perilaku yang besar dari jumlah yang ada.
3.		Generalization	Hubungan pada objek anak berbagi perilaku dan struktur data dari objek induk.
4.		Include	Memungkinkan 1 use case menggunakan fungsionalitas yang disediakan oleh use case yang lain.
5.		Extend	Memungkinkan 1 use case secara optimal menggunakan fungsilitas dari use case yang lain.
6.		Association	Menghubungkan antara objek satu ke objek yang lain.
7.		Aktor	Pengguna sistem yang berhubungan dengan sistem lain atau waktu



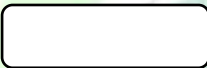


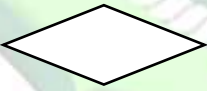

2.2.17 Activity Diagram

Activity Diagram merupakan gambaran perilaku alur kerja sistem untuk aktifitas, objek, state, transisi state, dan event. Activity Diagram berfokus

pada aktifitas yang terjadi dalam proses tunggal suatu sistem dan menunjukkan ketergantungan terhadap setiap aktifitasnya. [20]

Berikut adalah bentuk simbol yang digunakan dalam penggambaran Activity Diagram :

Tabel 2.4 Simbol *Activity Diagram*


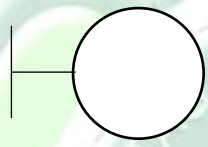




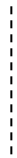
No	Simbol	Nama	Keterangan
1.		Start point	Merupakan awal dari sebuah aktivitas.
2.		End point	Merupakan Akhir dari sebuah aktivitas.
3.		Activities	Menggambarkan suatu proses atau kegiatan.
4.		Fork (percabangan)	Merupakan symbol yang digunakan untuk menggambarkan kegiatan yang dilakukan secara paralel menjadi satu.
5.		Join (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
6.		Decision point	Digunakan untuk menentukan pilihan <i>True or False</i> /Benar atau Salah.
7.		Swimlane	Digunakan untuk membagi activity diagram.

2.2.18 Sequence Diagram

Sequence Diagram merupakan diagram interaksi objek yang disusun berdasarkan urutan waktu yang memeberikan gambaran tahap demi tahap sesuai dengan skenario, termasuk kronologi kejadian yang seharusnya dilakukan secara logis untuk menghasilkan keluaran yang sesuai dengan Use Case Diagram. [21]

Berikut adalah bentuk simbol yang digunakan dalam penggambaran Sequence Diagram :

Tabel 2.5 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.		Entity class	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas yang membentuk gambaran sistem dan menjadi acuan penyusunan basis data.
2.		Boundary class	Merupakan kumpulan class yang menjadi interface atau interaksi antara satu actor dengan sistem seperti tampilan form.
3.		Control class	Merupakan objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab terhadap entitas.
4.		Message	Simbol mengirim pesan antar class.
5.		Recursive	Menggambarkan pengiriman pesan kepada dirinya sendiri.
6.		Activation	Mewakili sebuah eksekusi operasi dari objek, durasi aktivitas operasi berbanding lurus dengan panjang kotak.
7.		Lifeline	Garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

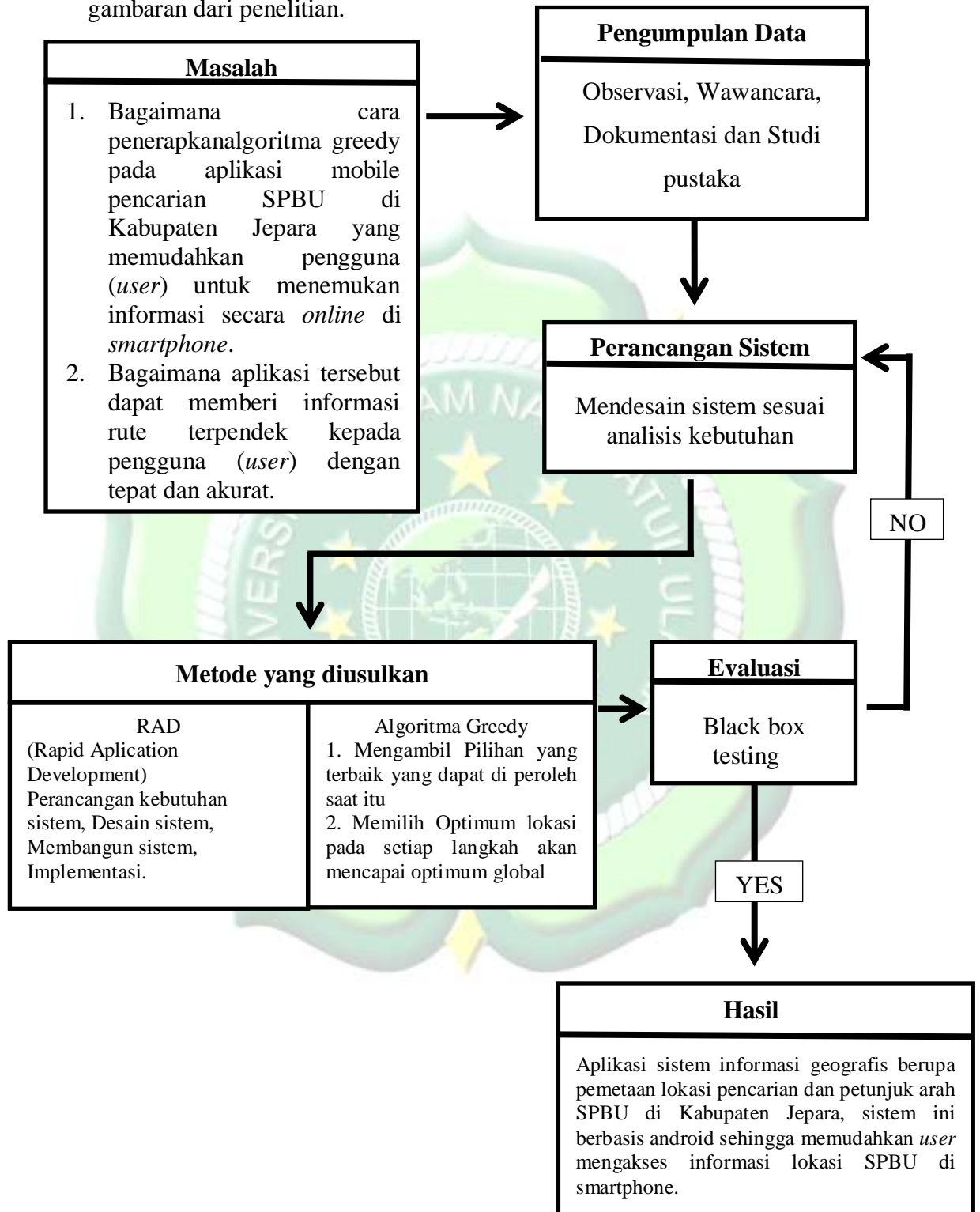
2.2.19 Black Box Testing

Blackbox Testing merupakan metode pengujian yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid [22]. Solusi peningkatan akurasi perlu dilakukan segera untuk memperbaiki celah error yang ditemukan, selanjutnya dilakukan pengujian keamanan secara intensif melalui jaringan internal (whitebox penetration testing) secara berkala oleh System Administrator atau Pengelola Sistem Informasi. [23]



2.3. Kerangka Pemikiran

Pada tahap ini peneliti membuat suatu kerangka pemikiran yang menjadi gambaran dari penelitian.



Gambar 2.6 Kerangka Pemikiran