

BAB II

LANDASAN TEORI

2.1 Pelayanan Administrasi Kelurahan

Pelayanan administrasi kelurahan merupakan bagian dari pelayanan publik yang diberikan pemerintah kepada masyarakat dalam bentuk pengurusan kebutuhan administratif, seperti surat keterangan usaha, surat keterangan tidak mampu, surat domisili, surat pengantar, serta dokumen pendukung lainnya. Dalam Undang-Undang Nomor 25 Tahun 2009, pelayanan publik dipahami sebagai kegiatan untuk memenuhi kebutuhan pelayanan masyarakat atas barang, jasa, dan/atau pelayanan administratif yang disediakan oleh penyelenggara pelayanan publik. Oleh karena itu, pelayanan administrasi kelurahan perlu dilaksanakan secara tertib, cepat, jelas, dan mudah diakses oleh masyarakat.

Kelurahan memiliki peran penting karena menjadi unit pemerintahan yang berhubungan langsung dengan kebutuhan administrasi warga. Pada tingkat kelurahan, masyarakat biasanya membutuhkan layanan yang sederhana, tetapi memiliki dampak besar bagi kehidupan sehari-hari, misalnya untuk keperluan pendidikan, bantuan sosial, usaha, pekerjaan, maupun layanan pemerintahan lainnya. Latupeirissa et al. (2024) menjelaskan bahwa digitalisasi pelayanan publik dapat membantu pemerintah daerah meningkatkan kualitas layanan, terutama ketika teknologi digunakan untuk memperbaiki akses, efisiensi, dan keteraturan proses pelayanan.

Dalam pelaksanaannya, pelayanan administrasi kelurahan masih sering menghadapi kendala apabila prosesnya dilakukan secara manual. Pencatatan data pemohon, pemeriksaan berkas, penyimpanan dokumen, dan pemantauan status pengajuan dapat menjadi kurang efektif apabila belum didukung sistem yang tertata. Dwimahendrawan et al. (2024) menunjukkan bahwa digitalisasi pelayanan administrasi kependudukan dapat mendekatkan layanan kepada masyarakat dan membantu proses pelayanan menjadi lebih praktis. Dengan demikian, penggunaan aplikasi mobile dalam pelayanan administrasi kelurahan dapat menjadi alternatif untuk membantu proses pengajuan surat, penyimpanan data, dan pemantauan layanan secara lebih efisien.

Berdasarkan uraian tersebut, pelayanan administrasi kelurahan dapat dipahami sebagai proses pemberian layanan administratif kepada masyarakat yang harus dilakukan secara jelas, mudah, dan tertata. Dalam penelitian ini, pelayanan administrasi kelurahan difokuskan pada pengembangan aplikasi mobile yang membantu pengajuan surat administrasi di Kelurahan Nusa Indah. Aplikasi ini diharapkan dapat mendukung proses pelayanan agar lebih praktis bagi masyarakat dan lebih tertata bagi petugas kelurahan.

2.2 Pelayanan Publik Digital

Pelayanan publik digital merupakan bentuk penyelenggaraan layanan pemerintah yang memanfaatkan teknologi informasi untuk mempermudah masyarakat dalam memperoleh layanan. Dalam konteks administrasi pemerintahan, digitalisasi tidak hanya berarti mengubah layanan manual menjadi layanan berbasis aplikasi, tetapi juga memperbaiki alur kerja agar pelayanan menjadi lebih cepat,

tertata, transparan, dan mudah diakses. Haug et al. (2024) menjelaskan bahwa transformasi digital di sektor publik dapat mendorong perubahan pada proses kerja, pola pelayanan, dan hasil layanan apabila diterapkan sesuai dengan kebutuhan organisasi serta masyarakat sebagai pengguna layanan.

Penerapan pelayanan publik digital menjadi penting karena masyarakat membutuhkan layanan yang sederhana, jelas, dan tidak terlalu bergantung pada proses tatap muka. Melalui sistem digital, proses pelayanan dapat dibuat lebih efisien, mulai dari pengisian data, pengajuan layanan, penyimpanan informasi, pemantauan status, hingga pengelolaan dokumen. Latupeirissa et al. (2024) menyatakan bahwa digitalisasi pelayanan publik dapat meningkatkan efisiensi, aksesibilitas, keterlibatan masyarakat, dan akuntabilitas pemerintah daerah. Namun, penerapannya tetap perlu memperhatikan kesiapan pengguna, ketersediaan perangkat, serta kemungkinan adanya kesenjangan digital bagi masyarakat yang belum terbiasa menggunakan teknologi.

Dalam ruang lingkup kelurahan, pelayanan publik digital dapat membantu memperbaiki proses administrasi yang sebelumnya masih dilakukan secara manual. Penggunaan aplikasi mobile, misalnya, memungkinkan masyarakat melakukan pengajuan surat, mengisi data diri, melampirkan dokumen pendukung, melihat riwayat pengajuan, dan memantau status layanan dengan lebih praktis. Bagi petugas kelurahan, sistem digital dapat membantu pencatatan data pemohon, pengelolaan riwayat pengajuan, dan penyusunan dokumen surat secara lebih tertata. Dengan demikian, pelayanan publik digital tidak hanya memberi kemudahan bagi

masyarakat, tetapi juga membantu aparat kelurahan dalam meningkatkan keteraturan proses kerja.

Pelayanan publik digital yang baik perlu dirancang berdasarkan kebutuhan pengguna. Aplikasi layanan pemerintah tidak cukup hanya dapat digunakan secara teknis, tetapi juga harus memiliki tampilan yang mudah dipahami, alur layanan yang jelas, serta fitur yang sesuai dengan kebutuhan masyarakat. Yuliantini dan Purnomo (2024) menjelaskan bahwa kualitas layanan mobile government berkaitan dengan konektivitas, interaktivitas, keaslian layanan, dan kemudahan pemahaman pengguna. Oleh karena itu, pengembangan aplikasi pelayanan administrasi kelurahan perlu memperhatikan kemudahan penggunaan agar aplikasi benar-benar membantu masyarakat, bukan justru menambah kesulitan baru.

2.3 Aplikasi Mobile

Aplikasi mobile merupakan perangkat lunak yang dirancang untuk digunakan pada perangkat bergerak, seperti smartphone dan tablet. Aplikasi ini memungkinkan pengguna melakukan berbagai aktivitas secara lebih praktis, seperti mengakses informasi, mengisi data, mengajukan layanan, menyimpan dokumen, dan memantau proses tertentu melalui perangkat yang mudah dibawa. Dalam kehidupan sehari-hari, aplikasi mobile menjadi semakin penting karena masyarakat dapat menggunakan layanan secara lebih fleksibel tanpa harus selalu datang langsung ke lokasi pelayanan.

Dalam konteks pelayanan publik, aplikasi mobile dapat menjadi media yang membantu pemerintah menyediakan layanan yang lebih mudah dijangkau oleh masyarakat. Melalui aplikasi mobile, proses administrasi dapat dibuat lebih

sederhana, mulai dari pengisian formulir, pengiriman data, penyimpanan informasi, pemantauan status layanan, hingga penyajian dokumen. Yuliantini dan Purnomo (2024) menjelaskan bahwa kualitas layanan *mobile government* berkaitan dengan konektivitas, interaktivitas, keaslian layanan, dan kemudahan pemahaman pengguna. Artinya, aplikasi pelayanan publik tidak hanya harus berjalan secara teknis, tetapi juga harus mampu memberikan pengalaman penggunaan yang jelas, mudah, dan dapat dipercaya oleh masyarakat.

Pengembangan aplikasi mobile perlu memperhatikan aspek kegunaan atau *usability*. Aplikasi yang baik seharusnya memiliki tampilan sederhana, menu yang mudah dipahami, alur penggunaan yang jelas, dan fungsi yang sesuai dengan kebutuhan pengguna. Weichbroth (2024) menjelaskan bahwa pengujian kegunaan aplikasi mobile perlu memperhatikan atribut, metode, dan alat evaluasi yang sesuai agar aplikasi benar-benar dapat digunakan secara efektif oleh pengguna. Dengan demikian, aplikasi pelayanan administrasi kelurahan perlu dirancang secara sederhana agar dapat digunakan oleh masyarakat dengan latar belakang kemampuan teknologi yang berbeda-beda.

Selain kemudahan penggunaan, aplikasi mobile juga perlu memperhatikan kualitas antarmuka, performa, dan pola interaksi. Weichbroth (2025) menunjukkan bahwa beberapa persoalan umum dalam aplikasi mobile berkaitan dengan arsitektur informasi, desain antarmuka, performa aplikasi, pola interaksi, dan tampilan visual. Jika aspek-aspek tersebut tidak diperhatikan, aplikasi dapat menjadi sulit digunakan meskipun fitur yang disediakan cukup lengkap. Oleh karena itu, pengembangan

aplikasi mobile pelayanan administrasi perlu menyeimbangkan kelengkapan fitur dengan kesederhanaan tampilan dan kejelasan alur layanan.

Pada penelitian ini, aplikasi mobile dikembangkan untuk mendukung pelayanan administrasi Kelurahan Nusa Indah. Aplikasi dirancang berbasis Android dengan fitur utama berupa autentikasi pengguna, dashboard layanan, profil pengguna, pengajuan surat, dokumen pendukung, manajemen data pengajuan, penyimpanan data lokal, status layanan, dokumen surat, dan bantuan penggunaan aplikasi. Fitur-fitur tersebut disusun untuk membantu masyarakat melakukan proses pengajuan surat secara lebih praktis, sekaligus membantu petugas kelurahan dalam mengelola data pengajuan secara lebih tertata.

2.4 Kodular

Kodular merupakan platform pengembangan aplikasi Android berbasis *visual block programming* yang memungkinkan pengguna membangun aplikasi dengan cara menyusun komponen dan blok logika program. Melalui Kodular, pengembang dapat merancang antarmuka aplikasi, menambahkan komponen, mengatur navigasi, menyusun proses input-output, serta menghubungkan aplikasi dengan beberapa komponen penyimpanan data. Kodular (2024) menjelaskan bahwa aplikasi dapat dibangun melalui kombinasi *components* dan *blocks*, sehingga pengembangan aplikasi tidak sepenuhnya bergantung pada penulisan kode secara manual.

Secara umum, Kodular dapat dipahami sebagai bagian dari pendekatan *low-code/no-code development*. Pendekatan ini memungkinkan proses pengembangan aplikasi dilakukan melalui antarmuka visual, komponen siap pakai, serta logika

berbasis blok. Matook et al. (2025) menjelaskan bahwa pengembangan berbasis *low-code* dapat membantu pengguna membangun solusi sistem informasi karena prosesnya lebih mudah dipelajari dan lebih dekat dengan pengalaman praktik langsung. Dengan karakteristik tersebut, Kodular dapat digunakan oleh pengembang pemula atau mahasiswa untuk membuat prototipe aplikasi yang memiliki kebutuhan sistem sederhana hingga menengah.

Kodular memiliki dua bagian utama yang penting dalam proses pengembangan aplikasi, yaitu bagian desain antarmuka dan bagian blok program. Bagian desain digunakan untuk menyusun tampilan aplikasi, seperti tombol, teks, formulir, gambar, menu, dan komponen lainnya. Sementara itu, bagian blok digunakan untuk mengatur logika aplikasi, misalnya proses login, validasi data, perpindahan halaman, penyimpanan data, pemanggilan data, dan pengolahan input pengguna. Dengan pola tersebut, proses pengembangan aplikasi menjadi lebih visual dan mudah dipahami karena alur kerja aplikasi dapat dilihat melalui susunan blok yang saling terhubung.

Dalam penelitian ini, Kodular digunakan untuk mengembangkan aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah. Aplikasi yang dikembangkan membutuhkan beberapa fitur utama, seperti autentikasi pengguna, dashboard layanan, profil pengguna, pengajuan surat, dokumen pendukung, manajemen data pengajuan, status layanan, dokumen surat, dan bantuan penggunaan aplikasi. Kodular dipilih karena mendukung pembuatan tampilan aplikasi yang sederhana, penggunaan komponen input data, pengaturan navigasi antarlayar, serta penyusunan logika aplikasi berbasis blok. Hal ini sesuai dengan

kebutuhan penelitian yang berfokus pada pembuatan prototipe aplikasi pelayanan administrasi berbasis Android.

Kelebihan Kodular terletak pada kemudahan penggunaannya. Pengembang dapat membuat aplikasi dengan cara memilih komponen, mengatur tampilan, dan menyusun blok perintah tanpa harus menulis kode dalam bahasa pemrograman yang kompleks. Kodular Companion juga memungkinkan aplikasi diuji secara langsung pada perangkat Android sebelum diekspor menjadi file APK. Dokumentasi Google Play untuk Kodular Companion menjelaskan bahwa fitur ini membantu pembuat aplikasi melakukan pengujian langsung tanpa harus mengekspor dan menyusun aplikasi terlebih dahulu. Kemudahan ini membuat proses pengembangan menjadi lebih cepat, terutama pada tahap perancangan dan pengujian awal aplikasi.

Selain kemudahan penggunaan, Kodular juga mendukung pengembangan aplikasi yang berorientasi pada kebutuhan pengguna. Dalam aplikasi pelayanan administrasi, tampilan dan alur layanan perlu dibuat sederhana agar masyarakat dapat memahami proses penggunaan aplikasi dengan mudah. Komponen visual yang tersedia pada Kodular dapat membantu pengembang membuat halaman login, formulir pengajuan surat, menu layanan, tampilan status pengajuan, dan halaman riwayat pengajuan. Dengan demikian, Kodular dapat mendukung pengembangan aplikasi yang tidak hanya berfungsi secara teknis, tetapi juga mudah digunakan oleh masyarakat.

Meskipun memiliki kelebihan, penggunaan Kodular juga memiliki keterbatasan. Aplikasi yang dikembangkan dengan pendekatan *low-code/no-code*

biasanya lebih sesuai untuk prototipe atau sistem dengan ruang lingkup terbatas. Triefflinger et al. (2025) menjelaskan bahwa penggunaan *low-code* perlu memperhatikan beberapa risiko, seperti tata kelola sistem, keamanan, skalabilitas, dan keterbatasan teknis. Oleh karena itu, aplikasi yang dikembangkan dalam penelitian ini diposisikan sebagai prototipe awal, bukan sebagai sistem final yang langsung terhubung dengan server pemerintahan, database kependudukan resmi, atau sistem administrasi terpusat.

Keterbatasan tersebut juga berhubungan dengan pengelolaan data. Kodular memang menyediakan komponen penyimpanan seperti TinyDB, tetapi penyimpanan tersebut lebih sesuai untuk kebutuhan data lokal pada aplikasi sederhana. Berdasarkan dokumentasi Kodular (2024), TinyDB merupakan komponen tidak terlihat yang digunakan untuk menyimpan data secara persisten di dalam aplikasi, sehingga data dapat digunakan kembali ketika aplikasi dijalankan. Dengan demikian, penggunaan Kodular dan TinyDB dalam penelitian ini perlu dipahami sebagai solusi awal untuk membangun aplikasi pelayanan administrasi yang ringan, sederhana, dan dapat diuji secara fungsional.

Kodular dalam penelitian ini berperan sebagai platform utama untuk merancang dan membangun aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah. Pemilihan Kodular didasarkan pada kemudahan pengembangan, dukungan visual block programming, ketersediaan komponen aplikasi, serta kesesuaiannya untuk membangun prototipe aplikasi Android. Dengan penggunaan Kodular, penelitian ini diharapkan dapat menghasilkan aplikasi yang mampu

membantu proses pengajuan surat, pengelolaan data pemohon, pemantauan status layanan, dan pembuatan dokumen surat secara lebih tertata.

2.5 TinyDB

TinyDB merupakan salah satu komponen penyimpanan data lokal yang tersedia pada platform Kodular. Komponen ini termasuk ke dalam *non-visible component*, yaitu komponen yang tidak ditampilkan secara langsung pada antarmuka aplikasi, tetapi bekerja di belakang sistem untuk menyimpan dan mengambil data. Dokumentasi Kodular (2024) menjelaskan bahwa TinyDB digunakan sebagai penyimpanan data persisten pada aplikasi, sehingga data yang telah disimpan tetap tersedia ketika aplikasi dijalankan kembali. Dengan demikian, TinyDB dapat membantu aplikasi menyimpan informasi sederhana tanpa harus menggunakan server atau database eksternal.

Dalam pengembangan aplikasi mobile, penyimpanan data memiliki peran penting karena aplikasi membutuhkan tempat untuk menyimpan informasi yang dimasukkan oleh pengguna. Data yang hanya disimpan dalam variabel biasa akan hilang ketika aplikasi ditutup atau dijalankan ulang. MIT App Inventor (2024) menjelaskan bahwa aplikasi akan melakukan inisialisasi ulang setiap kali dijalankan, sehingga nilai variabel tidak akan diingat kembali ketika pengguna menutup aplikasi. Berbeda dengan variabel biasa, TinyDB berfungsi sebagai penyimpanan data persisten, sehingga data tetap dapat dipanggil kembali ketika aplikasi digunakan pada waktu berikutnya.

TinyDB bekerja dengan konsep pasangan *tag* dan *value*. *Tag* berfungsi sebagai kunci atau penanda data, sedangkan *value* merupakan isi data yang

disimpan. Melalui mekanisme tersebut, aplikasi dapat menyimpan berbagai informasi sederhana, seperti nama pengguna, NIK, nomor telepon, jenis surat, nomor pengajuan, dan status layanan. Kodular (2024) menjelaskan bahwa TinyDB menyimpan data menggunakan *tag*, sehingga data dapat disimpan, dipanggil kembali, atau dihapus berdasarkan penanda tertentu. Cara kerja ini membuat TinyDB cukup mudah digunakan pada aplikasi yang tidak membutuhkan struktur basis data kompleks.

Pada penelitian ini, TinyDB digunakan sebagai media penyimpanan data lokal pada aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah. Data yang disimpan meliputi data login pengguna, profil pengguna, data pengajuan surat, dokumen pendukung, riwayat pengajuan, dan status layanan. Penggunaan TinyDB memungkinkan aplikasi menyimpan data pengajuan secara lokal pada perangkat, sehingga pengguna dapat melihat kembali riwayat pengajuan atau informasi tertentu tanpa harus mengisi ulang data dari awal. Dalam konteks prototipe, penyimpanan lokal seperti ini dapat membantu menunjukkan bagaimana aplikasi dapat mendukung pencatatan dan pengelolaan data administrasi secara sederhana.

Penggunaan TinyDB memiliki beberapa kelebihan dalam pengembangan aplikasi sederhana. Pertama, TinyDB mudah digunakan karena dapat dihubungkan langsung dengan blok program pada Kodular. Kedua, TinyDB tidak membutuhkan konfigurasi server, koneksi database online, atau pengaturan sistem penyimpanan yang rumit. Ketiga, penyimpanan dilakukan secara lokal pada perangkat, sehingga aplikasi tetap dapat menyimpan dan menampilkan data tertentu tanpa koneksi internet. Kodular (2024) dan MIT App Inventor (2024) sama-sama menjelaskan

bahwa TinyDB dapat digunakan sebagai penyimpanan data persisten, sehingga sesuai untuk kebutuhan penyimpanan sederhana seperti preferensi pengguna, data lokal, atau riwayat penggunaan aplikasi.

Meskipun demikian, TinyDB memiliki keterbatasan yang perlu diperhatikan. Karena data disimpan secara lokal pada perangkat, data tidak otomatis tersinkronisasi dengan perangkat lain atau server pusat. Apabila aplikasi dihapus atau data perangkat dibersihkan, data yang tersimpan dapat hilang. TinyDB juga kurang sesuai untuk sistem yang membutuhkan pengelolaan data dalam jumlah besar, akses banyak pengguna secara bersamaan, pencadangan otomatis, atau integrasi dengan database resmi. Dalam konteks pengembangan aplikasi berbasis *low-code*, Trieflinger et al. (2025) mengingatkan bahwa aspek skalabilitas, tata kelola, dan keterbatasan teknis perlu diperhatikan agar sistem tidak diklaim melebihi kemampuan platform yang digunakan.

Keterbatasan TinyDB juga berkaitan dengan aspek keamanan data. Aplikasi pelayanan administrasi kelurahan berpotensi menyimpan data pribadi masyarakat, seperti nama, NIK, alamat, nomor telepon, dan dokumen pendukung. Peruma et al. (2024) menjelaskan bahwa keamanan aplikasi mobile menjadi penting karena aplikasi sering mengelola data pribadi dan sensitif, sehingga pengembang perlu memperhatikan autentikasi, penyimpanan aman, izin akses, dan perlindungan privasi sejak tahap awal pengembangan. Oleh karena itu, penggunaan TinyDB dalam penelitian ini dibatasi pada kebutuhan prototipe dan pengujian aplikasi, bukan sebagai penyimpanan utama untuk sistem pelayanan administrasi resmi.

Dalam aplikasi yang dikembangkan, TinyDB berfungsi untuk mendukung beberapa proses utama. Pada fitur autentikasi, TinyDB dapat digunakan untuk menyimpan data login secara lokal. Pada fitur profil pengguna, TinyDB dapat menyimpan identitas pemohon agar dapat digunakan kembali pada pengajuan berikutnya. Pada fitur pengajuan surat, TinyDB dapat menyimpan data formulir, jenis surat, nomor pengajuan, dan status layanan. Sementara itu, pada fitur riwayat pengajuan, TinyDB dapat digunakan untuk menampilkan kembali data pengajuan yang pernah dibuat oleh pengguna. Dengan fungsi tersebut, TinyDB menjadi bagian penting dalam mendukung alur kerja aplikasi pelayanan administrasi berbasis mobile.

TinyDB dalam penelitian ini dipahami sebagai komponen penyimpanan data lokal yang digunakan untuk menyimpan dan mengambil kembali data aplikasi secara sederhana. Penggunaan TinyDB mendukung pengembangan aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah karena sesuai dengan kebutuhan prototipe yang ringan, praktis, dan tidak memerlukan server eksternal. Namun, karena sifatnya masih lokal, penggunaan TinyDB tetap perlu diberi batasan agar aplikasi tidak diklaim sebagai sistem final yang siap digunakan untuk pelayanan administrasi resmi secara penuh.

2.6 Metode Waterfall

Metode Waterfall merupakan salah satu model dalam *Software Development Life Cycle* (SDLC) yang digunakan untuk mengembangkan perangkat lunak secara bertahap dan berurutan. Model ini disebut Waterfall karena setiap tahapan pengembangan mengalir dari satu tahap ke tahap berikutnya, mulai dari analisis

kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan. Saravanos dan Curinga (2023) menjelaskan bahwa model Waterfall terdiri atas beberapa fase utama, yaitu *requirements analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, serta *operation and maintenance*. Dengan pola tersebut, Waterfall menekankan keteraturan proses dan kejelasan keluaran pada setiap tahap pengembangan.

Dalam pengembangan perangkat lunak, Waterfall digunakan ketika kebutuhan sistem relatif dapat ditentukan sejak awal. Model ini cocok untuk proyek yang memiliki ruang lingkup jelas, fitur yang dapat dirancang sebelum implementasi, dan proses pengembangan yang tidak membutuhkan perubahan besar secara terus-menerus. Sommerville (2016) menjelaskan bahwa model Waterfall menggambarkan proses perangkat lunak yang direncanakan dan terdokumentasi, sehingga setiap tahap menghasilkan dokumen atau keluaran yang menjadi dasar bagi tahap berikutnya. Oleh karena itu, Waterfall sering digunakan dalam penelitian pengembangan aplikasi karena alurnya mudah dipahami, sistematis, dan sesuai untuk pembuatan prototipe dengan kebutuhan yang telah dirumuskan.

Tahap pertama dalam metode Waterfall adalah analisis kebutuhan (*requirement analysis*). Pada tahap ini, pengembang mengidentifikasi masalah, mengumpulkan data, dan menentukan kebutuhan sistem yang akan dikembangkan. Dalam penelitian ini, analisis kebutuhan dilakukan untuk mengetahui proses pelayanan administrasi di Kelurahan Nusa Indah, jenis surat yang dibutuhkan, data yang harus dimasukkan, dokumen pendukung yang diperlukan, serta fitur aplikasi

yang sesuai dengan kebutuhan pengguna. Tahap ini penting karena kesalahan dalam memahami kebutuhan dapat memengaruhi kualitas rancangan dan implementasi sistem pada tahap berikutnya.

Tahap kedua adalah perancangan sistem (*system design*). Pada tahap ini, kebutuhan sistem yang telah diperoleh diterjemahkan ke dalam rancangan aplikasi. Perancangan mencakup alur kerja sistem, struktur menu, rancangan antarmuka, alur penyimpanan data, serta hubungan antarfitur dalam aplikasi. Pressman dan Maxim (2020) menjelaskan bahwa tahap desain dalam rekayasa perangkat lunak berfungsi untuk menerjemahkan kebutuhan pengguna menjadi rancangan teknis yang dapat diimplementasikan. Dalam penelitian ini, tahap perancangan meliputi pembuatan flowchart, use case diagram, rancangan tampilan aplikasi, dan rancangan penggunaan TinyDB sebagai media penyimpanan data lokal.

Tahap ketiga adalah implementasi (*implementation*). Pada tahap ini, rancangan yang telah dibuat diwujudkan menjadi aplikasi yang dapat dijalankan. Implementasi dilakukan dengan membangun fitur-fitur aplikasi sesuai dengan rancangan, seperti login pengguna, dashboard layanan, profil pengguna, pengajuan surat, dokumen pendukung, manajemen data pengajuan, status layanan, dan dokumen surat. Dalam penelitian ini, implementasi dilakukan menggunakan Kodular sebagai platform pengembangan aplikasi Android berbasis *visual block programming*. TinyDB digunakan untuk menyimpan data secara lokal, seperti data login, profil pengguna, data pengajuan surat, riwayat pengajuan, dan status layanan.

Tahap keempat adalah pengujian (*testing*). Pengujian dilakukan untuk memastikan bahwa aplikasi berjalan sesuai dengan kebutuhan dan rancangan yang

telah dibuat. Pada model Waterfall, pengujian dilakukan setelah tahap implementasi selesai, sehingga fungsi-fungsi aplikasi dapat diperiksa secara menyeluruh. Saravanos dan Curinga (2023) menyebutkan bahwa salah satu fase Waterfall adalah *integration and system testing*, yaitu tahap untuk memastikan bagian-bagian sistem dapat bekerja secara bersama-sama. Dalam penelitian ini, pengujian dilakukan menggunakan metode *black-box testing*, yaitu pengujian yang berfokus pada input dan output aplikasi tanpa melihat struktur kode internal.

Tahap kelima adalah pemeliharaan atau evaluasi sistem (*maintenance*). Pada tahap ini, aplikasi yang telah diuji dapat diperbaiki apabila ditemukan kesalahan atau ketidaksesuaian fungsi. Pemeliharaan juga dapat mencakup penyesuaian aplikasi berdasarkan masukan pengguna, perbaikan tampilan, atau penyempurnaan alur sistem. Dalam penelitian ini, tahap pemeliharaan lebih difokuskan pada evaluasi hasil pengujian dan perbaikan prototipe. Hal ini dilakukan karena aplikasi yang dikembangkan masih berada pada tahap prototipe awal dan belum menjadi sistem final yang digunakan secara resmi dalam pelayanan administrasi kelurahan.

Metode Waterfall memiliki beberapa kelebihan. Pertama, alur pengembangan jelas karena setiap tahap dilakukan secara berurutan. Kedua, dokumentasi sistem lebih mudah disusun karena setiap tahap menghasilkan keluaran tertentu, seperti dokumen kebutuhan, rancangan sistem, aplikasi, dan hasil pengujian. Ketiga, model ini sesuai digunakan untuk penelitian skripsi yang memiliki waktu terbatas dan kebutuhan sistem yang telah ditentukan sejak awal. Tjahjanto et al. (2022) menunjukkan bahwa Waterfall banyak digunakan dalam pengembangan sistem

informasi karena prosesnya terstruktur dan membantu pengembang menyusun tahapan kerja secara sistematis.

Meskipun demikian, metode Waterfall juga memiliki keterbatasan. Karena prosesnya cenderung berurutan, perubahan kebutuhan yang muncul di tengah proses pengembangan dapat menjadi sulit diterapkan apabila tahap sebelumnya sudah selesai. Selain itu, kesalahan yang terjadi pada tahap awal dapat berdampak pada tahap berikutnya apabila tidak diidentifikasi dengan baik. Saravanos (2025) menjelaskan bahwa berbagai penelitian simulasi Waterfall menunjukkan pentingnya pelaporan model yang jelas, penggunaan tahapan yang terdefinisi, dan pengelolaan proses agar hasil pengembangan lebih dapat ditelusuri. Oleh karena itu, penggunaan Waterfall dalam penelitian ini perlu dilakukan secara hati-hati, terutama pada tahap analisis kebutuhan dan perancangan sistem.

Metode Waterfall dalam penelitian ini digunakan sebagai alur pengembangan aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah. Model ini dipilih karena sesuai dengan karakteristik penelitian yang memiliki kebutuhan sistem relatif jelas, ruang lingkup aplikasi terbatas, dan tahapan pengembangan yang dapat disusun secara sistematis. Melalui metode Waterfall, proses penelitian dilakukan mulai dari analisis kebutuhan, perancangan sistem, implementasi menggunakan Kodular dan TinyDB, pengujian menggunakan *black-box testing*, hingga evaluasi hasil pengujian. Dengan demikian, pengembangan aplikasi diharapkan dapat berjalan lebih terarah, terdokumentasi, dan sesuai dengan tujuan penelitian.

2.7 Black-Box Testing

Black-box testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsi sistem berdasarkan input dan output, tanpa melihat struktur kode program di dalamnya. Dalam pengujian ini, penguji hanya memperhatikan apakah fitur aplikasi berjalan sesuai dengan kebutuhan dan menghasilkan keluaran yang benar ketika diberikan masukan tertentu. Saravanos dan Curinga (2023) menjelaskan bahwa pengujian sistem dalam proses pengembangan perangkat lunak bertujuan memastikan bahwa bagian-bagian sistem dapat bekerja sesuai fungsi yang telah dirancang. Dengan demikian, *black-box testing* sesuai digunakan untuk menguji aplikasi dari sudut pandang pengguna.

Pengujian *black-box* dilakukan dengan menyusun skenario pengujian berdasarkan fungsi-fungsi utama aplikasi. Setiap skenario pengujian berisi fitur yang diuji, data masukan, langkah pengujian, hasil yang diharapkan, dan hasil pengujian yang diperoleh. Dalam pendekatan ini, penguji tidak perlu mengetahui bagaimana logika program disusun, tetapi cukup memeriksa apakah aplikasi memberikan respons yang sesuai. Sultansyah et al. (2025) menggunakan *black-box testing* untuk menguji fitur permohonan informasi publik pada website pemerintah, sehingga metode ini relevan untuk sistem layanan publik yang berfokus pada ketepatan fungsi layanan.

Dalam penelitian ini, *black-box testing* digunakan untuk menguji fungsionalitas aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah. Pengujian dilakukan pada fitur-fitur utama aplikasi, seperti login pengguna, logout, validasi data login, dashboard layanan, input dan edit profil pengguna, pengajuan surat, validasi kelengkapan formulir, penyimpanan data menggunakan TinyDB,

riwayat pengajuan, status layanan, preview dokumen, serta export surat ke format PDF. Setiap fitur diuji untuk memastikan bahwa input yang diberikan pengguna menghasilkan output yang sesuai dengan rancangan aplikasi.

Metode *black-box testing* memiliki beberapa teknik yang dapat digunakan, antara lain *equivalence partitioning*, *boundary value analysis*, *decision table testing*, dan *state transition testing*. *Equivalence partitioning* digunakan untuk membagi data uji ke dalam kelompok data valid dan tidak valid. *Boundary value analysis* digunakan untuk menguji nilai batas, misalnya panjang NIK, jumlah karakter nomor telepon, atau kelengkapan formulir. *Decision table testing* dapat digunakan ketika sistem memiliki beberapa kondisi keputusan, seperti pengajuan diterima apabila formulir lengkap dan dokumen pendukung tersedia. Sementara itu, *state transition testing* dapat digunakan untuk memeriksa perubahan status pengajuan, misalnya dari “diproses” menjadi “disetujui” atau “ditolak”. Salim (2024) menunjukkan bahwa *state transition testing* dapat digunakan dalam pengujian *black-box* untuk memeriksa perubahan kondisi pada sistem berbasis web.

Kelebihan *black-box testing* adalah pengujian dapat dilakukan dari sudut pandang pengguna akhir. Hal ini penting karena aplikasi pelayanan administrasi kelurahan harus mudah digunakan dan memberikan hasil sesuai kebutuhan masyarakat. Maspupah dan Ramayanti (2024) menjelaskan bahwa *black-box testing* efektif digunakan untuk pengujian fungsional karena penguji dapat menilai apakah sistem sudah memenuhi kebutuhan tanpa harus memeriksa kode program. Dengan demikian, metode ini cocok digunakan dalam penelitian pengembangan aplikasi yang menekankan kesesuaian fitur dengan kebutuhan pengguna.

Meskipun demikian, *black-box testing* juga memiliki keterbatasan. Karena pengujian hanya berfokus pada fungsi luar sistem, metode ini tidak dapat mengetahui secara langsung kesalahan pada struktur kode, efisiensi algoritma, atau keamanan internal aplikasi. Casper et al. (2024) menjelaskan bahwa pengujian berbasis akses *black-box* memiliki keterbatasan karena evaluator hanya dapat melihat respons sistem dari luar, tanpa mengetahui mekanisme internal yang menghasilkan respons tersebut. Oleh karena itu, dalam penelitian ini *black-box testing* digunakan untuk menguji fungsionalitas aplikasi, sedangkan pengujian keamanan tingkat lanjut, analisis kode, dan pengujian performa tidak menjadi fokus utama penelitian.

Black-box testing dalam penelitian ini digunakan sebagai metode pengujian fungsional untuk memastikan bahwa aplikasi mobile pelayanan administrasi Kelurahan Nusa Indah berjalan sesuai rancangan. Pengujian ini dilakukan dengan memeriksa kesesuaian antara input yang diberikan, proses yang dijalankan, dan output yang dihasilkan oleh aplikasi. Hasil pengujian kemudian digunakan untuk mengetahui apakah aplikasi telah memenuhi kebutuhan dasar pengguna dan apakah fitur-fitur utama sudah dapat digunakan secara baik.