

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Dalam melaksanakan penelitian, kami meninjau penelitian yang terdahulu yang berkaitan dengan penelitian kami sebagai acuan dalam pembuatan, tinjauan penelitian sebagai berikut:

Penelitian berikut ini yang berjudul “Perancangan Aplikasi Wedding Organizer berbasis android” diteliti oleh Haris Ginanjar, dan Ridwan Setiawan tahun 2020. Permasalahan yang terjadi di penelitian ini adalah perkembangan teknologi telah merambah diberbagai bidang, salah satunya adalah bidang perdagangan dengan adanya perdagangan e-marketplace. Peneliti memanfaatkan kemajuan teknologi dan menerapkan ke bidang jasa Wedding Organizer. Dengan merancang aplikasi Wedding Organizer peneliti dapat memudahkan calon pengantin dalam memesan Wedding Organizer. Peneliti menggunakan metode Rational Unified Process. [5]

Penelitian selanjutnya diteliti oleh Fernanda Yerisha, Hartinah Ridwan dengan judul ”Pengembangan Aplikasi Katalog Wedding Organizer Berbasis Android Menggunakan Restful Web Service Di Ruang Lingkup Kota Malang” pada tahun 2019. Latar belakang penelitian tersebut karena di kota Malang banyak calon mempelai sulit mengarut waktu untuk mencari vendor maupun Wedding Organizer yang sesuai dengan keinginan mereka. Mereka cukup kesulitan dalam mencari informasi vendor maupun Wedding Organizer. Sebab itu dibuat penelitian ini sebagai jembatan antara pihak vendor /Wedding Organizer dengan calon mempelai. Peneliti menggunakan metode restful web service dalam daya dalam bentuk JSON yang ada di API [2]

Penelitian selanjutnya yaitu diteliti oleh Aldianto Lobo dengan judul “Perancangan Sistem Informasi Penyewaan Peralatan Pernikahan berbasis android (studi kasus: UD Manalagi Kabupaten Toraja Utara) pada tahun 2019”. Penelitian ini dilatarbelakangi oleh perusahaan UD Manalagi Dagang yang membutuhkan sebuah sistem informasi yang dapat mempertemukan antara

pihak pengelola dan penyewa dalam hal penyewaan aksesoris melalui android karena UD tersebut belum memanfaatkan media informasi data tersebut. Pengelolaan data di UD tersebut ditulis dalam buku pemesanan serta ditulis dalam buku pemesanan yang menyebabkan karyawan kesulitan membuat laporan penyewaan yang dibutuhkan oleh pimpinan UD Manalagi kabupaten Toraja Utara. [6]

2.2 Tinjauan Pustaka

2.2.1 Definisi Wedding Organizer

Wedding Organizer adalah jasa yang membantu calon pengantin membuat perencanaan pelaksanaan serangkaian acara pernikahan. Dan tentu saja mereka merencanakan pesta pernikahan yang sesuai dengan kantong masing-masing calon pengantin, Wedding Organizer dibutuhkan karena memudahkan calon pengantin dalam merencanakan pesta pernikahan yang sakral, dengan dibantu oleh pekerja yang profesional dari Wedding Organizer sehingga menghasilkan pesta pernikahan yang sesuai dengan idaman masing-masing pasangan. [5]

2.2.2 Mahkota Wedding Jepara

Mahkota Wedding Adalah sebuah jasa persewaan dekorasi dan rias pernikahan ber alamatkan di jalan Sunan Mantingan no 14 dema'an Jepara. Mulai beroperasi pada tahun 2017.



Gambar 2. 1 Peta Denah Lokasi Mahkota Wedding

Gambar di atas merupakan letak mahkota wedding yang diambil dari google map.

2.2.3 Definisi Android

Android adalah sebuah sistem operasi perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android merupakan sistem informasi yang menyediakan platform terbuka (Open Source) bagi para pengembang untuk menciptakan aplikasi mereka. [7]

2.2.4 Definisi android studio

Android Studio merupakan sebuah IDE (Integrated Development Environment) untuk pengembangan aplikasi android, aplikasi ini dipublikasikan oleh Google pada tanggal 16 mei 2013 dan tersedia secara gratis dibawah lisensi Apache 2.0, Android studio ini menggantikan software pengembangan android sebelumnya yaitu Eclipse [8]

2.2.5 Android SDK (Software Development Kit)

Android SDK adalah tools bagi para programmer yang ingin mengembangkan aplikasi berbasis Google Android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode, dan tutorial [9].

2.2.6 Database

Perancangan basis data bertujuan untuk menggambarkan hubungan antardata, atribut dan data yang akan disimpan, dan kebutuhan ruang penyimpanan. Perancangan basis data dibuat untuk mendapatkan struktur yang dapat mempermudah dan mempercepat dalam mengakses dan memanipulasi data serta bebas dari redudansi data. Hasil perancangan basis data harus dapat mengakomodir kebutuhan saat ini dan masa depan [10]

2.2.7 Firebase database real time

Firestore database real time database merupakan basis data online yang dapat digunakan sebagai media penyimpanan data dari aplikasi. Data disimpan dalam bentuk JSON dan dapat disinkronkan secara real time ke setiap client yang terhubung. Layanan ini memiliki 3 kemampuan yaitu real time, offline, accessible from client device. [10]Firestore memiliki beberapa fitur, diantaranya adalah realtime database yang disimpan secara cloud, layanan ini menggunakan Application program interface (API), data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung, apabila ada perubahan pada data yang tersimpan, maka setiap user yang terhubung akan menerima pembaruan data secara otomatis [11]

2.2.8 Unified Modeling Language (UML)

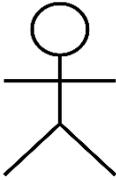
Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standar digunakan untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan UML kita dapat membuat semua jenis aplikasi piranti lunak dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi, dan jaringan apapun. UML dapat diterapkan pada berbagai bahasa pemrograman. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya maka UML lebih cocok digunakan untuk pemrograman dalam bahasa berorientasi objek [12]

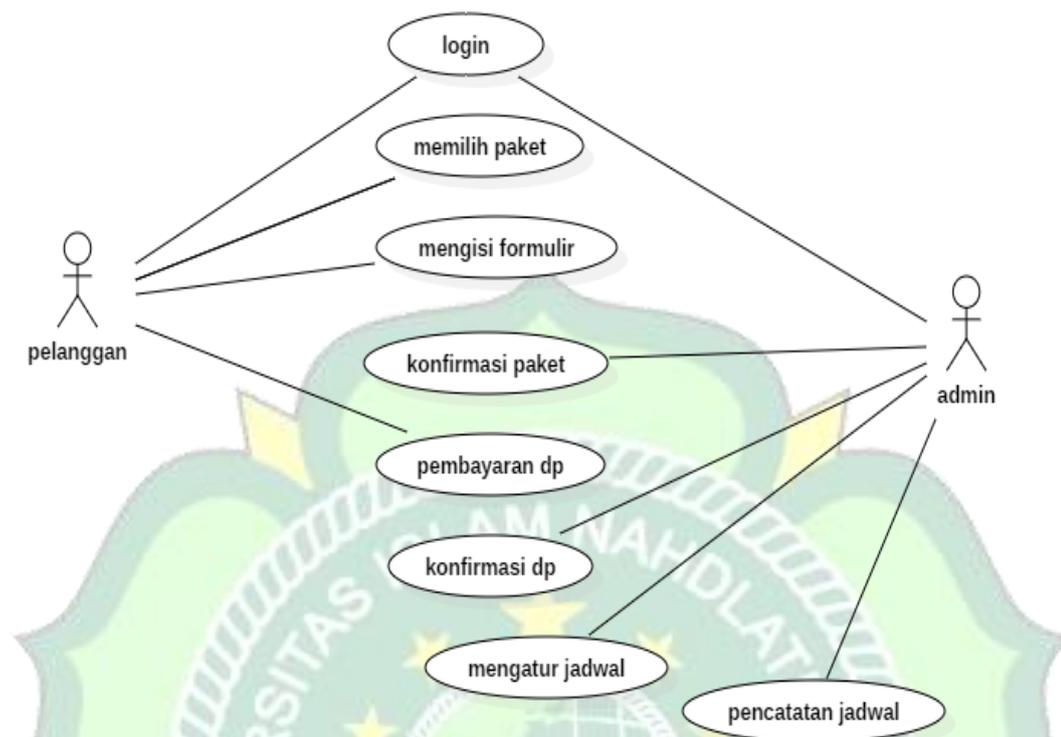
1. Definisi Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (user). Sehingga pembuatan use case diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah Use Case Diagram mempresentasikan sebuah interaksi antara aktor dengan sistem [12]

Tabel 2. 1 Use Case Diagram

No	Gambar	Nama	Keterangan
----	--------	------	------------

1.		Aktor	Pengguna sistem yang berhubungan dengan sistem lain atau waktu
2.		Include	Memungkinkan 1 use case menggunakan fungsionalitas yang disediakan oleh use case yang lain
3.		Generalization	Hubungan pada objek anak berbagi perilaku dan struktur data dari objek induk.
4.		Extend	Memungkinkan 1 use case secara optimal menggunakan fungsi dari use case yang lain
5.		Use Case	deskripsi dari urutan aksi yang ditampilkan sistem.
6.		Collaboration	interaksi elemen lain yang berkerja sama untuk menyediakan perilaku yang besar dari jumlah yang ada.
7.		Association	Menghubungkan antara objek satu ke objek yang lainnya



Gambar 2. 2 Contoh Use Case Diagram

Ini adalah salah satu gambar Use Case yang akan saya gunakan pada penelitian ini yaitu use case diagram. Gambar di atas terdapat dua aktor (pengguna). Pengguna pada aplikasi ini akses yang bisa digunakan pas pengguna yaitu:

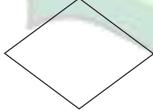
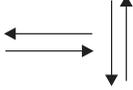
1. Pembeli : fungsi aplikasi yang bisa dilakukan sebagai pembeli jika sesuai urutan dari awal. Pengguna harus login ke aplikasi dulu apabila sudah memilih paket yang disediakan pada aplikasi tersebut dan terakhir melakukan pembayaran yang dilakukan pada sewa tersebut
2. Admin : Fungsi aplikasi sebagai admin dengan login dulu kemudian melihat paket yang akan disewa dan mengkonfirmasi apakah paket tersebut bisa disewa atau tidak, jika sudah admin juga bisa melakukan konfirmasi pembayaran apabila pelanggan sudah melakukan

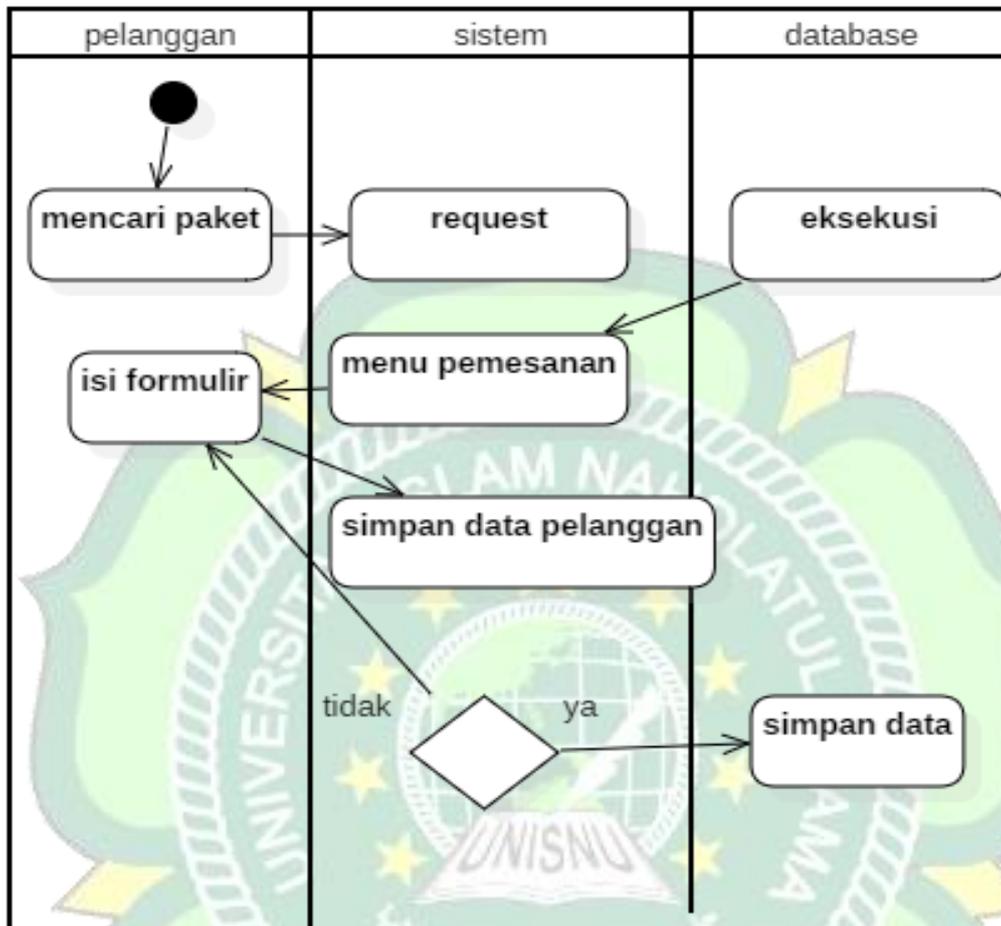
pembayaran.

1. Definisi Activity Diagram

Activity diagram digunakan untuk menggambarkan aliran kerja (aktivitas dari sebuah sistem atau proses bisnis). Dimana diagram aktivitas menggambarkan aktivitas sistem bukan yang di lakukan actor, jadi aktivitas dapat dilakukan oleh sistem simbol yang di gunakan activity diagram adalah sebagai berikut [12]

Tabel 2. 2 Diagram Activity

No.	Gambar	Nama	Keterangan
1.		Activity	Memperlihatkan Bagaimana Masing-Masing Kelas Antarmuka Saling Berinteraksi Satu Sama Lain
2.		Action	State Dari Sistem Yang Mencerminkan Eksekusi Dari Suatu Aksi
3.		Initial Node	Bagaimana Objek Terbentuk Atau Diawali
4.		Activity Final Node	Bagaimana Objek Dibentuk Dan Diakhiri
5.		Decision	Digunakan Untuk Menggambarkan Suatu Tindakan Yang Harus Diambil Pada Kondisi Tertentu
6.		Line Connector	Digunakan Untuk Menghubungkan Simbol Satu Dengan Simbol Lainnya



Gambar 2. 3 Contoh Activity Diagram

- *Pelanggan memilih paket
- *Sistem merespon dengan menampilkan menu pemesanan
- *Pelanggan memilih paket sistem merespon memproses dan menampilkan form data pelanggan
- *Pelanggan mengisi data
- * Sistem merespon dengan memverifikasi apakah data sudah benar atau belum jika sudah benar, data tersebut disimpan di database

2. Definisi Sequence Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Berfungsi untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. [12]

Tabel 2. 3 Sequence Diagram

No	Simbol	Nama	Deskripsi
1		Start State	Titik awal siklus hidup suatu elemen
2		State	Menggunakan kondisi suatu elemen
3		Final State	Titik Akhir Yang Menjadi Kondisi Suatu Elemen
4		Action State	Menggambarkan Keadaan Elemen Dalam Suatu Aliran Aktifitas
5		Flow Control	Menggambarkan Aliran Aktifitas Dari Satu Elemen Ke Elemen Lain

3. Definisi Class Diagram

Class Diagram adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class Diagram menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi). [12]

Tabel 2. 4 Class Diagram

No	Nama	Simbol	Deskripsi
----	------	--------	-----------

1	Asosiasi		Hubungan Statis Antar Class, Umumnya Menggambarkan Class Yang Memiliki Atribut Berupa Class Lain.
2	Agregasi		Hubungan Yang Menyatakan Bagian (“Terdiri Atas”) Atau Bisa Disebut Relasi.
3	Composition		Sebuah Class Tidak Biasa Berdiri Sendiri Merupakan Bagian Dari Class Lain, Maka Class Tersebut Memiliki Relasi Composition.
4	Dependency		Hubungan Yang Menunjukkan Operasi Pada Suatu Yang Menggunakan Class Lain.

2.2.9 Pengujian perangkat lunak

Pengujian perangkat lunak merupakan sebuah proses pengujian program yang dimaksudkan untuk mencari kesalahan pada software. Pengujian juga bertujuan untuk memastikan bahwa software memiliki kualitas yang baik. Kualitas software yang baik adalah software memenuhi kriteria yang diinginkan dan memberikan produktivitas yang tinggi.

Pengujian perangkat lunak memiliki peranan penting dalam suatu sistem informasi, dengan pengujian ini dapat diketahui galat atau error yang akan muncul pada perangkat lunak. Dengan pengujian perangkat lunak diharapkan dapat meminimalisir kesalahan dan cacat pada sebuah software dan sebagai pengukuran kualitas dari software tersebut [13]

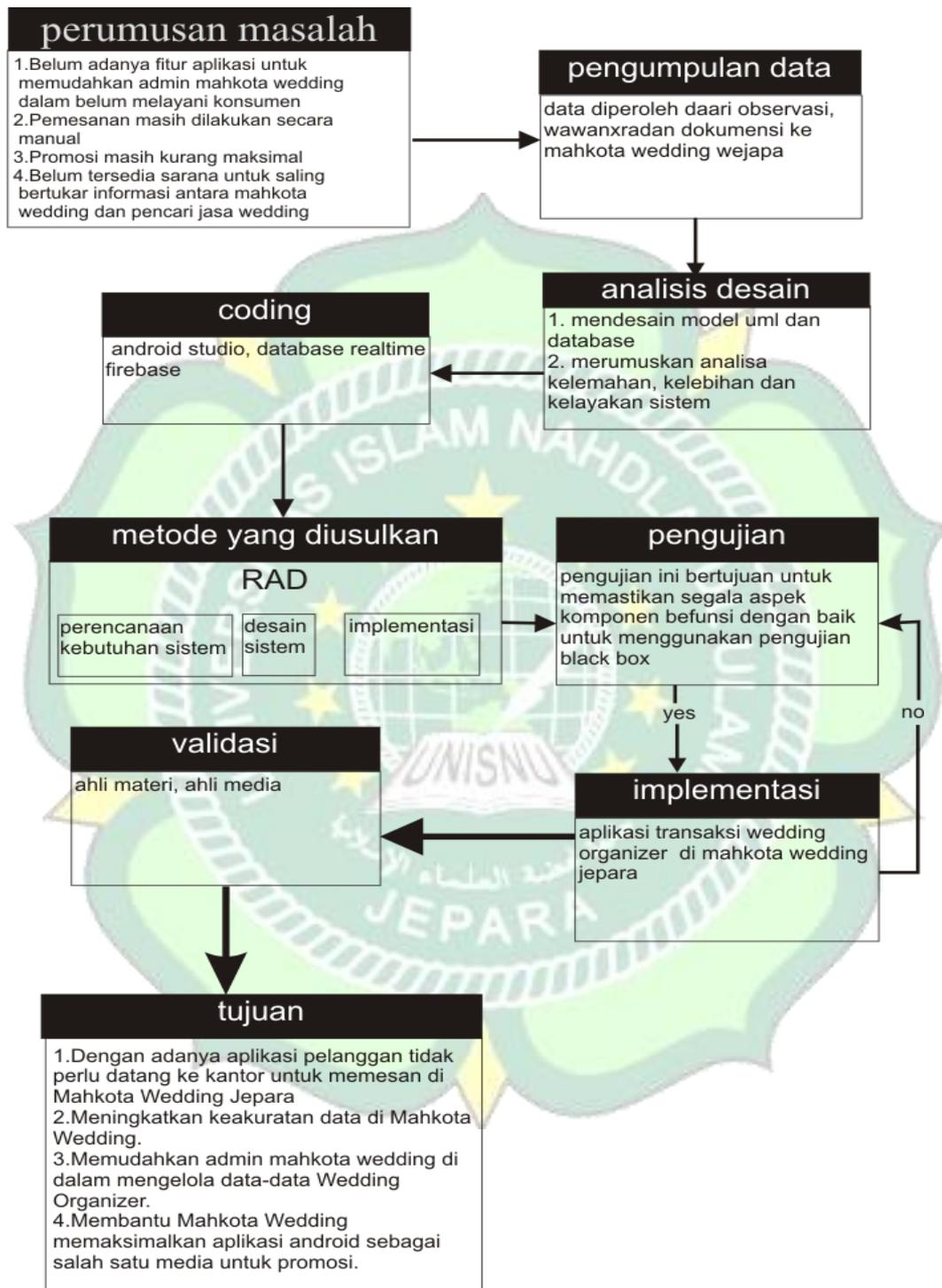
2.2.10 Black box testing

Black-Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Black-box Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. Black-box Testing memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program [14]

Keuntungan menggunakan metode Black Box Testing adalah penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan inkonsistensi dalam spesifikasi persyaratan. Programmer dan tester keduanya saling bergantung satu sama lain. [15]

Kekurangan dari metode Black box Testing adalah uji kasus sulit disain tanpa spesifikasi yang jelas. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer. Beberapa bagian back tidak diuji sama sekali. [15]

2.3 Kerangka Pemikiran



Gambar 2. 4 Kerangka Pemikiran