

BAB 2

LANDASAN TEORI

2.1 Aplikasi

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 komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Pengertian Aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, Aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user. Pengertian Aplikasi menurut para ahli:

1. Pengertian Aplikasi menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, intruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga computer dapat memproses *input* menjadi *output*.
2. Pengertian Aplikasi menurut Kamus Besar Bahasa Indonesia (1998:52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program computer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.
3. Menurut Rachmad Hakim S., Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows & permainan (game), dan sebagainya.
4. Menurut Harip Santoso, Aplikasi adalah suatu kelompok *file* (*form, class, report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed asset*, dll. [2]

2.2 Wedding Organizer

2.2.1 Pengertian Wedding Organizer

Wedding Organizer adalah suatu jasa khusus yang membantu calon pengantin dan keluarga dalam perencanaan dan supervise pelaksanaan rangkaian acara pesta pernikahan sesuai dengan jadwal yang telah ditetapkan. *Wedding Organizer* juga mengelola *event* pernikahan dari mulai Akad nikah, Upacara adat, Pelaminan, *Cathering*, rias pengantin, dokumentasi dan lain-lain.

2.2.2 Mengapa dan kapan membutuhkan jasa *Wedding Organizer*

Inilah beberapa alasan kapan saat yang tepat membutuhkan jasa *Wedding Organizer*:

1. Waktu yang Sangat Berharga

Khususnya bila calon pengantin atau keluarga sibuk terkait dengan aktifitas pekerjaan yang tinggi sehingga sulit menyisakan waktu yang cukup untuk menyiapkan sendiri segala perencanaan dan perlengkapan acara.

2. Efisiensi waktu dan tenaga

Begitu banyak macam kebutuhan sebuah pesta pernikahan dan tersedia beraneka ragam pilihan. Bila belum memiliki sendiri data atau pengalaman menggunakan suatu jasa/produk, sungguh melelahkan bila harus mencari dan membandingkannya sendiri satu persatu. Dengan memanfaatkan semua informasi mengenai pernikahan yang disediakan oleh seorang *Wedding Organizer*, kita dapat menghemat waktu dan tenaga.

3. Tanggung jawab profesional atas kelancaran acara

Menjelang pesta, ditengah kegembiraan dan kesibukan dalam mempersiapkan penampilan diri secara sempurna, hamper tidak mungkin lagi untuk memeriksa sendiri kesiapan perlengkapan pesta seperti dekorasi, *Cathering*, Fotografer, dan lain-lain. Dengan kontrak kerja yang profesional, *Wedding Organizer* akan bertanggung jawab secara penuh atas kelancara acara.

4. Penampilan yang sempurna

Pesta pernikahan akan menjadi kenangan seumur hidup. Kesiapan fisik dan mental yang sempurna adalah kunci dari segalanya. Kerjasama yang terpadu pengguna jasa dan sebuah tim yang professional akan membantu mewujudkannya.

2.2.3 Komponen-Komponen Sistem WO

Dibawah ini ada beberapa bahasan komponen yang penting untuk kita ketahui, antara lain:

1. Tenda

Tenda adalah salah satu perlengkapan utama dalam perencanaan sebuah acara/pesta. Keberadaannya adalah sebuah keharusan yang tidak boleh tidak, karena merupakan kebutuhan utama dalam penyelenggara sebuah acara/pesta. Tenda juga berguna untuk menutupi tempat acara pernikahan dari panas atau hujan. Tenda terdiri dari tiang-tiang yang terbuat dari besi dan lapisan atas dilapisi oleh terpal dan dibagian bawah terpal dilapisi oleh kain untuk menghias tenda yang disebut balon.

2. Baju pengantin

Baju pengantin menambah keindahan momen pernikahan. Momen pernikahan akan selalu dikenang kedua mempelai yang mengikat janji suci. Baju pengantin indah nan memukau merupakan salah satu elemen pernikahan yang tak hanya menampilkan keindahan, tapi juga gengsi sang pengantin.

3. Riasan

Komponen riasan adalah suatu kegiatan mengubah penampilan dari bentuk asli sebenarnya dengan bantuan bahan dan alat kosmetik. Istilah *make up* lebih sering ditujukan kepada perubahan bentuk wajah, meskipun sebenarnya seluruh tubuh bisa di hias. Riasan dapat berguna bukan untuk menghias wajah tetapi juga untuk membuat wajah mempelai wanita lebih cerah dihari pernikahannya.

4. Upacara Adat

Upacara pernikahan adalah upacara adat yang diselenggarakan dalam rangka menyambut peristiwa pernikahan. Kedua mempelai meyakini bahwa dengan mengikuti upacara adat ini maka mereka dapat membuat acara pernikahan menjadi lebih khidmat. Di Indonesia upacara pernikahan dilakukan dengan dua cara, tradisional dan modern. Upacara pernikahan secara tradisional dilakukan menurut aturan-aturan adat setempat. Upacara pernikahan modern dilakukan dengan mengikuti aturan-aturan dari luar negeri. Biasanya gaya yang dipakai adalah gaya eropa. Pernikahan yang dilakukan dengan aturan islam mungkin dapat juga dimasukkan ke dalam kategori upacara pernikahan modern.

5. Hiburan

Komponen hiburan berguna untuk menghibur para tamu undangan yang hadir memberikan selamat serta doa restu kepada kedua mempelai, sehingga tidak membuat suatu acara pernikahan menjadi membosankan. Hiburan pada resepsi pernikahan akan membawa kedalam Suasana Kegembiraan, romantisme penuh kenangan. Dengan pengisi hiburan musik dan lagu mampu menciptakan atmosfer keindahan dalam kesyahduan pesta perkawinan.

6. Katering

Pada komponen ini akan membahas katering yang merupakan suatu komponen yang menyediakan makanan kepada para tamu undangan yang hadir pada acara pernikahan tersebut. masakan dan minuman dapat dipilih dan tentukan untuk acara dengan variasi penyajian yang semuanya akan menjadikan pesta pernikahan semakin berkesan.

7. Pelaminan

Komponen pelaminan ini merupakan suatu komponen yang menghiasi dekorasi pelaminan tempat untuk kedua mempelai, biasanya dalam pelaminan ada hiasan air terjun dibelakang tempat duduk kedua mempelai dan juga biasanya dibuat sebuah taman kecil yang terdiri dari pot-pot kecil didepan tempat duduk kedua mempelai agar dekorasi pelaminan terlihat lebih indah.

8. Dokumentasi

Dokumentasi adalah kegiatan untuk merekam dan menyimpan berbagai data penting yang dihasilkan dari suatu kegiatan. Acara seindah pernikahan perlu sebuah dokumentasi, karena momen ini sangat berharga yang hanya ada sekali seumur hidup. Untuk itu dokumentasi sangat diperlukan.

9. Akad Nikah

Ini merupakan komponen yang terakhir tapi komponen yang paling penting dalam sebuah acara pernikahan. Akad nikah merupakan ikrar pernikahan. Yang dimaksud dengan akad nikah ialah ijab daripada pihak wali perempuan atau wakilnya dan qabul dari pihak calon suami atau wakilnya. Akad nikah merupakan syarat wajib dalam proses atau upacara perkawinan menurut Islam. Akad nikah boleh dijalankan oleh wali atau diwakilkan kepada seorang juru nikah.

2.2.4 Tahapan Sistem Wedding Organizer

1. Perencanaan Pernikahan

Calon pengantin datang ke *Wedding Organizer* untuk melakukan perencanaan pernikahan termasuk memilih paket pernikahan sesuai dengan tema yang mereka inginkan. Pemilihan paket pernikahan yaitu melakukan pilihan untuk alat-alat yang akan digunakan dalam acara pernikahan nanti, seperti memilih Gaun/Baju pengantin, ukuran tenda dan jenisnya, upacara adat yang akan dilaksanakan, dan lain-lain. Setelah pengantin melakukan pemilihan alat-alat pengantin, calon pengantin dan *Wedding Organizer* menentukan besaran harga.

2. Melakukan Persiapan

Pada tahap ini *Wedding Organizer* akan melakukan persiapan untuk menyusun acara calon mempelai. Tahap mempersiapkannya yaitu dimulai dari melakukan organisasi dari pihak keluarga atau kerabat mempelai untuk membantu menjadi pelengkap diacara pernikahan seperti pagar ayu, penerima tamu undangan, dan lain-lain. Lalu *Wedding Organizer* melakukan pemasangan tenda, balon tenda,

dinding kain, tempat pelaminan serta mempersiapkan *Cathering* kurang lebih H-3.

3. Acara Pernikahan

Setelah melalui tahap pertama dan kedua dengan matang tibalah pada tahap selanjutnya yaitu acara pernikahan. Dalam acara pernikahan ini pertama kali yang dilakukan adalah melakukan rias wajah dan pemasangan pakaian calon mempelai, pagar ayu, dan lain-lain. Setelah melalui tahap tersebut maka kedua pengantin akan melakukan akad nikah untuk mengikat janji pernikahan atau ijab qabul. Menuju ketahap selanjutnya pasangan yang telah resmi menjadi suami istri tersebut akan melakukan upacara adat menurut daerah sekitar, atau pun daerah asal yang telah ditentukan.

Selama acara berlangsung dari awal hingga akhir *Wedding Organizer* akan merekam melalui gambar maupun video untuk mendokumentasikan momen yang indah tersebut. Tidak ketinggalan juga acara hiburan yang akan membawa mempelai dan tamu undangan kedalam suasana yang menyenangkan.

Sampailah kedua mempelai pada tahap penerimaan tamu diatas pelaminan. Yaitu melakukan salam-salaman dan foto bersama kepada pengunjung yang datang dan menyambut para tamu dengan santapan *Cathering* yang telah disediakan sebelumnya. Proses tersebut berlangsung terus hingga acara selesai sesuai dengan waktu yang telah ditentukan.

4. Akhir Acara

Setelah semua tahapan berjalan dengan lancar, ini adalah tahapan terakhir dimana *Wedding Organizer* akan melakukan pelepasan tenda dan dekorasi, pencucian kain-kain dekorasi dan baju-baju pengantin, dan melakukan pelunasan pembayaran.

2.2.5 Ruang Lingkup Wedding Organizer

Ruang lingkup pekerjaan seorang *Wedding Organizer* adalah :

1. Memberikan input kepada calon pengantin mengenai hal-hal yang harus diperhatikan dalam perencanaan dan pelaksanaan upacara pernikahan.
2. Mencari lokasi resepsi.
3. Menyusun *budget*.
4. Membantu perencanaan mengenai tema, alur, dan dekorasi pesta.
5. Membuat buku program acara pernikahan.
6. Mengkoordinasikan dan mengarahkan *Job Description* panitia keluarga.
7. Fasilitasi, negosiasi dan koorniasi dengan pihak gedung/hotel dan *supplier/vendor* seperti *Cathering*, dekorasi, fotografer, perias, grup music, dan lain-lain.
8. Pengurusan persyaratan akad nikah dan perizinan lain-lain.
9. Menyussun jadwal kerja dan jadwal pembayaran.
10. Mengatur *setting* ruangan dan *flow* tamu di rumah maupun di tempat resepsi.
11. *Supervisi* pelaksanaan upacara pernikahan agar segala sesuatunya dapat berjalan dengan baik sesuai dengan rencana.
12. Mengambil langkah-langkah pengamanan bila terjadi keadaan darurat (sebatas dengan kewenangan yang diberikan).

2.3 Java

Java adalah sebuah bahasa pemrograman yang diciptakan oleh James Gosling, seorang developer dari Sun Microsystem pada tahun 1991. Selanjutnya Java dikembangkan Sun Microsystem dan banyak digunakan untuk menciptakan *Executable Content* yang dapat didistribusikan melalui *network*.

Java adalah bahasa pemrograman *Object-Oriented* dengan unsur-unsur seperti bahasa C++ dan bahasa-bahasa lainnya yang memiliki *libraries* yang cocok untuk lingkungan internet. Java dapat melakukan banyak hal dalam melakukan pemrograman, seperti membuat animasi halaman *web*, pemrograman Java untuk Ponsel dan aplikasi interaktif.

Penelitian ini menggunakan java dikarenakan memakai bahasa pemrograman java dengan menggunakan *tools* Android Studio, maka dari itu penelitian ini memerlukan landasan teori tentang java.

2.3.1 Karakteristik-Karakteristik Java

1. Sederhana

Bahasa pemrograman Java menggunakan sintaks yang mirip dengan bahasa C++ namun sintaks pada Java telah banyak diperbaiki, terutama dengan menghilangkan *pointer* yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *garbage collection*.

2. Berorientasi Objek

Java merupakan bahasa pemrograman berorientasi objek yang memungkinkan program untuk dibuat secara modular dan digunakan kembali.

3. Terdistribusi

Java dibuat untuk memudahkan distribusi aplikasi dengan adanya *networking libraries* yang terintegrasi dalam Java.

4. *Interpreted*

Program Java dijalankan menggunakan program Interpreter, yaitu Java Virtual Machine (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *bytecodes* dapat dijalankan pada berbagai *platform*.

5. Robust

Java mempunyai reliabilitas yang tinggi. Kompiler pada Java mempunyai kemampuan mendeteksi *error* yang lebih baik dibandingkan bahasa pemrograman yang lain. Java mempunyai *Runtime Exception Handling* untuk membantu mengatasi *error* pada pemrograman.

6. *Secure*

Sebagai bahasa pemrograman aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga agar aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

7. *Architecture Neutral*

Program Java tidak bergantung pada *platform* dimana program akan dijalankan. Cukup dibuat satu program yang dapat dijalankan pada berbagai *platform* dengan Java Virtual Machine.

8. *Portable*

Source code maupun program Java dapat dengan mudah dibawa ke berbagai *platform* berbeda tanpa harus dikompilasi ulang.

9. *Performance*

Kinerja Java sering kali dikatakan kurang, namun kinerja Java dapat ditingkatkan menggunakan *compiler* Java lain seperti buatan Inprise, Microsoft maupun Symantec yang menggunakan Just In Time Compilers (JIT).

10. *Multithreaded*

Java dapat membuat suatu program yang mampu melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. *Dynamic*

Java dapat didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan suatu *class* dengan menambahkan *properties* ataupun metode dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut. [3]

2.4 Android

Android merupakan sistem operasi yang memang khusus dirancang untuk smartphone dan tablet. Sistem android ini memiliki basis Linux yang mana dijadikan sebagai pondasi dasar dari sistem operasi android. Linux sendiri merupakan sistem operasi yang memang khusus dirancang untuk komputer. Android memang dirancang untuk dipasang pada perangkat-perangkat mobile touchscreen (smartphone dan tablet). Sehingga sistem operasi yang berada di dalam smartphone saat ini memang menyesuaikan dari spesifikasi kelas low-end hingga high-end. Sehingga perkembangan sistem android memang cukup meningkat tajam. Android merupakan sistem operasi yang terbuka (open source) yang mana berarti jika pihak Google memperbolehkan dan

membebaskan bagi pihak manapun untuk dapat mengembangkan sistem operasi tersebut. Bahkan anda sendiri pun juga dapat mengembangkan sistem android yang memang sesuai dengan keinginan anda.

Sistem Android memiliki gudang aplikasi dan game yaitu *Google Play Store*, yang mana disini anda bisa mendownload serta menggunakan aplikasi atau game yang terdapat di *Google Play Store* sepuasnya dengan menggunakan perangkat seluler dengan sistem android. Uniknya, Android menggunakan nama-nama makanan untuk membedakan versi sistem android yang diluncurkannya. Android menggunakan huruf depan dari nama makanan tersebut sebagai penanda peningkatan versi sistemnya. Mulai dari Cupcake Android 1.5 (C), Donuts Android 1.6 (D), Éclair Android 2.0-2.1 (E) atau Marshmallow Android 6.0 (M) hingga yang terbaru sekarang versi Oreo Android 8.0 (O).

Maka dari itu penelitian pembangunan aplikasi ini membahas tentang android karena menggunakan *platform* berbasis *mobile*, dan banyak juga orang yang menggunakan sistem operasi berbasis android. Pembangunan aplikasi ini pun cocok diterapkan dalam android.

2.4.1 Sejarah Android

Perjalanan Android dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, Nick Sears dan Chris White mendirikan Android.Inc, di California US. Visi Android untuk mewujudkan *mobile device* yang lebih peka dan mengerti pemilikinya, kemudian menarik raksasa dunia maya *Google*. *Google* kemudian mengakuisisi Android pada Agustus 2005. OS Android dibangun berbasis *platform* Linux yang bersifat *open source*, senada dengan Linux, Android juga bersifat *Open Source*. Dengan nama besar *Google* dan konsep *open source* pada OS Android, tidak membutuhkan waktu lama bagi android untuk bersaing dan menyisihkan Mobile OS lainnya seperti Symbian, Windos Mobile, Blackberry dan iOS. Kini siapa yang tak kenal Android yang telah menjelma menjadi penguasa *Operating System* bagi

Smartphone. Android telah terinstal pada sebagian besar semua sistem operasi dan berikut ini beberapa versi Android: [4]

1. Android 1.5 (*Cupcake*)

Cupcake dirilis 30 April 2009. *Cupcake* menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 *Cupcake* adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

2. Android 1.6 (*Donut*)

Android V1.6, *codename Donut*, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan *reboot*, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin. Bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan topping di atasnya.

3. Android 2.0/2.1 (*Eclair*)

Android 2.0/2.1 *Eclair* Dirilis 26 Oktober 2009. *Eclair* adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan coklat di atasnya.

4. Android 2.2 (*Froyo: Frozen Yoghurt*)

Dirilis 20 Mei 2010. Menggunakan *codename Froyo*, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari Yoghurt. *Froyo* singkatan dari *Frozen Yoghurt*, *Froyo* adalah yoghurt yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim.

5. Android 2.3 (*Gingerbread*)

Android versi 2.3 *Gingerbread* dirilis resmi tanggal 6 Desember 2010. *Gingerbread* merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. Android 3.0 (*Honeycomb*)

Dirilis tanggal 22 February 2011. *Honeycomb* adalah sereal sarapan manis yang sudah dibuat rahun 1965 oleh Posting Sereal. Seperti namanya, *Honeycomb*/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. Android 4.0 (*ICS: Ice Cream Sandwich*)

Android 4.0-4.0.2 API Level 14 dan 4.0.3-4.0.4 API Level 15 pertama dirilis 19 Oktober 2011. Dinamai *Ice Cream Sandwich*. *Ice Cream Sandwich* adalah lapisan es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. Android 4.1 (*Jelly Bean*)

Android *Jelly Bean* diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel 3.0.31*. Terdiri dari Android 4.1 API Level 16, Android 4.2 API Level 17, Android 4.3 API Level 18. Penamaan *Jelly Bean* mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. Android 4.4 (*Kitkat*)

Android 4.4 Kitkat API level 19. Google mengumumkan Android KitKat (dinamai dengan izin dari Nestle dan Hershey) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. Kitkat merupakan merk sebuah coklat yang dikeluarkan oleh Nestle. Rilis berikutnya setelah nama KitKat diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai '*Key Lime Pie*'.

10. Android 5.0 (*Lollipop*)

Pembaruan yang mencolok pada *Lollipop* tampak dari sisi desainnya yang diperhalus dan disesuaikan dengan zaman. Selain itu, fitur-fitur yang sudah hadir pada Android sebelumnya ditingkatkan. Inovasi kurang terasa pada versi ini. Satu-satunya yang lumayan baru adalah dukungan untuk gambar berformat RAW. Format itu memungkinkan para ilustrator, fotografer atau *graphic designer* untuk menyimpan file dengan ukuran besar agar bisa di edit tanpa mengurangi kualitas.

11. Android 6.0 (*Marshmallow*)

Rilis terbaru Android 6.0 bernama *Marshmallow*. Pintasan terpintar dari sini ke sana, *Now on Tap* mengantisipasi apa yang Anda butuhkan saat ini. Dengan ketukan sederhana, Anda bisa mendapatkan kartu dengan informasi dan aplikasi yang berguna yang perlu Anda ketahui. Tidak perlu lagi mengurus baterai dari aplikasi yang jarang digunakan, Aplikasi Siaga membatasi pengaruhnya terhadap masa pakai baterai sehingga biaya Anda bertahan lebih lama. Tidak perlu memberikan akses aplikasi sepanjang waktu. Android Marshmallow memungkinkan Anda menentukan apa yang ingin Anda bagikan dan kapan. Matikan izin juga kapan saja. Sidik jari Anda membuka kunci ponsel dan memungkinkan Anda secara aman masuk dan keluar di Android Pay, Play Store, dan bahkan di aplikasi. [4]

12. Android 7.0 (*Nougat*)

Di perkenalkan pada tahun 2016, os android versi terbaru android yang di beri nama Android *Nougat* ini merupakan salah satu versi android termutakhir yang di kembangkan oleh google. Terjadi pembaharuan pada sisi os tersebut dengan penambahan fitur google *assistant* yang berfungsi sebagai pengganti dari google *now* yang sebelumnya tersedia di os android versi lebih lawas lainnya. Selain beberapa fitur utama diatas,ada juga penambahan 1500 emoji terbaru, yang hobi chat pakai *emoji* dan multi Bahasa, android *Nougat* merupakan salah satu pilihan terbaik karena selain menambah 75 emoji baru, os ini juga memiliki kemampuan 2 bahasa dalam waktu bersamaan. Jika di os android versi sebelumnya anda bisa membuka beberapa aplikasi langsung dan berpindah dari satu aplikasi ke aplikasi lain dengan fitur tab, di android *Nougat* anda tidak perlu berpindah atau *multitasking*, karena os ini sudah mendukung 2 aplikasi android langsung berjalan dengan waktu bersamaan.

13. Android 8.0 (*Oreo*)

Lebih pintar, lebih cepat, lebih kuat, dan lebih manis dari sebelumnya. Dua kali lebih cepat mulai dari tugas favorit Anda lebih cepat dengan dua kali kecepatan

boot saat menyalakan. Batas latar belakang Android Oreo membantu meminimalkan aktivitas latar belakang di aplikasi yang paling jarang Anda gunakan, ini adalah kekuatan super yang bahkan tidak dapat Anda lihat. Isi Otomatis mengingat login Anda untuk membawa Anda ke aplikasi favorit Anda dengan kecepatan supersonik. Android Oreo mengenali teks saat Anda mengetuk atau memilih, kemudian menyarankan langkah logis berikutnya melalui aplikasi yang disarankan. Memungkinkan Anda melihat dua aplikasi sekaligus, seperti memiliki kekuatan super dan visi laser. Bicara tentang penghemat hidup baik Anda berbicara, bermain, bekerja atau streaming, Anda dapat merasa yakin tentang menjaga baterai Anda kuat dan penuh kehidupan. [6]

2.4.2 Kelebihan Android

Banyak sekali *platform* untuk perangkat selular saat ini, namun ada beberapa hal yang menjadi kelebihan Android diantaranya:

1. *User Friendly*, yang dimaksudkan disini adalah sistem Android sangat mudah untuk dijalankan. Sama halnya pada sistem operasi Windows yang ada pada komputer. Bagi yang belum terlalu terbiasa dengan *smartphone*, hanya membutuhkan waktu sebentar saja untuk bisa mempelajari sistem Android.
2. Akan sangat mudah mendapat beragam notifikasi dari *smartphone* yang dimiliki. Untuk mendapatkannya, bisa mengatur beberapa akun yang anda miliki seperti SMS, Email, *Voice Dial*, dan lainnya.
3. Keunggulan lainnya terdapat dari segi tampilan sistem Android yang menarik dan tidak kalah baiknya dengan iOS (Apple). Hal ini dikarenakan dari awal, Android memang mengusung konsep dan teknologi iOS hanya saja Android merupakan versi murah dari iOS.
4. Sistem operasi ini memang memiliki konsep *open source* yang mana pengguna dapat bebas mengembangkan sistem android versi miliknya sendiri. Sehingga akan banyak sekali *Custom ROM* yang bisa digunakan.

5. Tersedia beragam pilihan aplikasi menarik, bahkan hingga jutaan aplikasi. Dari mulai aplikasi gratis hingga aplikasi berbayar. Dapat mendownloadnya langsung di *Google Playstore* yang tersedia pada smartphone android.

2.4.3 Kekurangan Android

Adapun kekurangan dari android sendiri diantaranya:

1. *Update system* yang kurang efektif. Sistem Android memang seringkali mengalami peningkatan versi yang ditawarkan kepada penggunanya. Namun untuk melakukan *update system* Android ini bukanlah hal yang mudah. Diharuskan untuk menunggu masing-masing vendor merilis resmi *update* terbaru dari sistem android tersebut. Dan bagi beberapa pengguna, hal ini kurang begitu menyenangkan.
2. Baterai yang cepat habis, mungkin hal ini sering dialami oleh pengguna sistem Android saat ini. Apalagi jika sering menyalakan paket data serta menggunakan *widget* dan aplikasi yang berjalan terlalu berlebihan sehingga menyebabkan daya baterai berkurang dengan cepat. Untuk mengatasi hal tersebut, bisa mengurangi aktivitas yang ada pada *smartphone*.
3. Sering mengalami lambat, hal ini biasanya berkaitan dengan spesifikasi dari masing-masing perangkat seluler. Namun meskipun begitu, jika sistem android memang tidak bersahabat dengan aplikasi-aplikasi yang anda miliki tentu saja akan berdampak pada respon yang lama pada penggunaan *smartphone* pengguna. Hal ini dikaitkan dengan RAM atau prosesor yang memang kurang memadai. Sehingga akan lebih baik jika menyimpan aplikasi yang memang sesuai dengan *smartphone* yang dimiliki.

2.5 Firebase

Firebase merupakan sebuah layanan infrastruktur *backend-as-a-service* (BaaS) yang diakuisisi oleh Google pada Oktober 2014 silam. Firebase menawarkan kemudahan kepada para pengembang perangkat lunak dalam membangun aplikasi

yang lebih baik serta mengembangkan bisnis yang sukses melalui seluruh fitur komplementernya. Saat ini, Firebase sudah memiliki fitur *Analytics*, *Cloud Messaging*, *Authentication*, *Realtime Database*, *Storage*, *Hosting*, *Test Lab*, *Crash Reporting*, *Notifications*, *Remote Config*, *App Indexing*, *Dynamic Links*, *Invites*, *AdWords*, dan *AdMob*. Seluruh fitur tersebut dikemas dalam sebuah SDK Firebase tunggal sehingga dengan kemudahan yang ditawarkan para pengembang perangkat lunak dapat fokus untuk memecahkan masalah *customer* melalui perangkat lunak yang dibuatnya dan tidak menghabiskan banyak waktu dalam membangun infrastruktur yang kompleks.

2.5.1 Firebase Realtime Database

Firestore *Realtime Database* merupakan sebuah layanan NoSQL *cloud-hosted database* yang dimiliki oleh Firebase SDK. Layanan ini menawarkan penyimpanan data yang dapat disinkronisasikan secara *realtime* terhadap seluruh *client* yang terhubung. Layanan ini memiliki 3 kemampuan inti yaitu *realtime*, *offline* dan *accessible from client devices*. Maksud dari *realtime* adalah jika terdapat perubahan data pada *database*, maka seluruh *client* yang terhubung akan secara otomatis mendapatkan perubahannya dalam hitungan milidetik. Kemudian *offline*, yaitu aplikasi yang menggunakan Firestore *Realtime Database* akan tetap responsif bahkan saat *offline*. Hal ini disebabkan karena Firebase SDK dapat mempertahankan data dan perubahannya pada media penyimpanan *client*. Pada saat *client* terhubung ke jaringan internet, maka Firebase SDK akan melakukan penyesuaian otomatis atas catatan perubahan data yang disimpan pada media penyimpanan *client* dengan kondisi terkini dari Firestore *server*. Kemampuan inti yang terakhir adalah *accessible from client devices*. Layanan ini menawarkan kemudahan untuk mengakses Firestore *Realtime Database* secara langsung dari sebuah perangkat *mobile* atau sebuah peramban *web* tanpa membutuhkan *server application*. [8]

Kesimpulan dari penjelasan diatas adalah firebase dapat digunakan dalam penelitian pembangunan aplikasi ini. Dalam pembangunan aplikasi ini memanfaatkan fitur Firestore *Realtime Database*, bahwa fitur ini terdapat perubahan data pada

database yang membuat user dapat melakukan *update* data secara *real time* kapan saja pada saat *online* maupun *offline*.

2.6 SQLite

SQLite merupakan sebuah *system management* basis data relasional yang bersifat ACID - *compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. *SQLite* merupakan proyek yang bersifat *public domain* yang dikerjakan oleh D. Richard Hipp.

SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada *SQL-92*, termasuk transaksi yang bersifat *atomic*, konsistensi basis data, isolasi, dan durabilitas (dalam bahasa Inggris lebih sering disebut ACID), *trigger*, dan kueri-kueri yang kompleks. [9]

Alasan pemakaian *SQLite* dalam pembangunan aplikasi ini adalah digunakan untuk penyimpanan data di aplikasinya sendiri, dimana penyimpanan tersebut terdapat keranjang pesanan. Dalam melakukan pesanan oleh pelanggan akan langsung disimpan ke dalam keranjang pesanan sehingga diperlukan *SQLite* dalam penyimpanannya.

2.7 Object Oriented Program (OOP)

Object Oriented Program (OOP) merupakan paradigma baru dalam rekayasa *software* yang didasarkan pada obyek dan kelas. (Ronald J.N., 1996). Diakui para ahli bahwa *objectoriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa *software*. Object-oriented memandang *software* bagian per bagian dan menggambarkan satu bagian tersebut dalam satu obyek.

Teknologi obyek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh obyek. Dengan demikian keunggulan teknologi obyek adalah bahwa model yang dibuat akan sangat mendekati dunia nyata yang masalahnya akan dipecahkan oleh sistem yang dibangun. Model obyek, atribut dan perlakuannya bisa langsung diambil dari obyek yang ada di dunia nyata.

Ada 4 (empat) prinsip dasar dari pemrograman berorientasi obyek yang menjadi dasar kemunculan UML, yaitu abstraksi, enkapsulasi, modularitas dan hirarki. Berikut dijelaskan satu persatu secara singkat: [5]

1. Abstraksi memfokuskan perhatian pada karakteristik obyek yang paling penting dan paling dominan yang bisa digunakan untuk membedakan obyek tersebut dari obyek lainnya.
2. Enkapsulasi menyembunyikan banyak hal yang terdapat dalam obyek yang tidak perlu diketahui oleh obyek lain. Dalam praktek pemrograman, enkapsulasi diwujudkan dengan membuat suatu kelas *interface* yang akan dipanggil oleh obyek lain, sementara didalam obyek yang dipanggil terdapat kelas lain yang mengimplementasikan apa yang terdapat dalam kelas *interface*.
3. Modularitas membagi sistem yang rumit menjadi bagian-bagian yang lebih kecil yang bisa mempermudah developer memahami dan mengelola obyek tersebut.
4. Hirarki berhubungan dengan abstraksi dan modularitas, yaitu pembagian berdasarkan urutan dan pengelompokkan tertentu. Misalnya untuk menentukan obyek mana yang berada pada kelompok yang sama, obyek mana yang merupakan komponen dari obyek yang memiliki hirarki lebih tinggi. Semakin rendah hirarki obyek berarti semakin jauh abstraksi dilakukan terhadap suatu obyek.

Kesimpulan yang didapat dari pembahasan diatas adalah dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah, tidak dapat melihat bagaimana cara menyelesaikan suatu masalah tersebut, tetapi objek-objek apa yang bisa melakukan pemecahan masalah yang dibahas.

2.8 *Unified Modeling Language (UML)*

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, et. al. 2004).

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Management Group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rational Software Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rational.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal* (RFP) yang dikeluarkan oleh OMG (Braun, et.al. 2001). Kemudian pada Januari 1997 IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies dan Softeam juga menanggapi Request for Proposal (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung. Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga saat ini UML versi terbaru adalah versi 2.0.

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. [5]

Berdasarkan pembahasan diatas dapat disimpulkan bahwa UML bagian dari *Object Oriented Analysis* yang menitikberatkan analisis dari sisi pengguna atau orang-orang diluar sistem, bukan menitikberatkan pada aliran data di sistem.

2.8.1 Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”* Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009): [5]

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

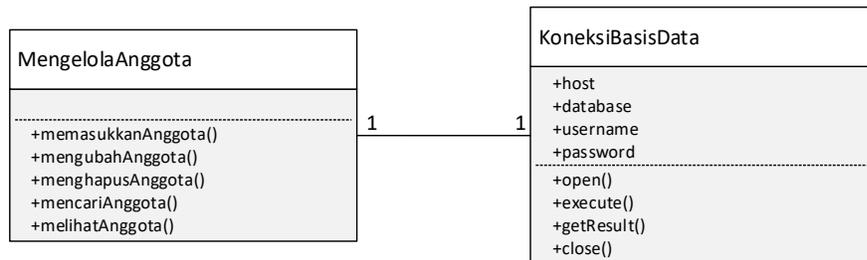
2.8.2 Diagram UML

Diagram UML, diantaranya adalah :

1. Class Diagram

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. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di

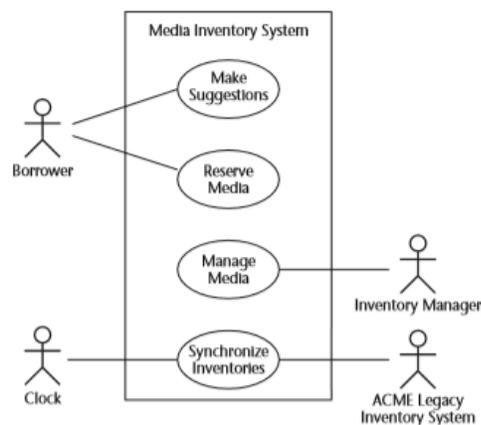
dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai. [7]



Gambar 2. 1 Notasi Class Diagram

2. Usecase Diagram

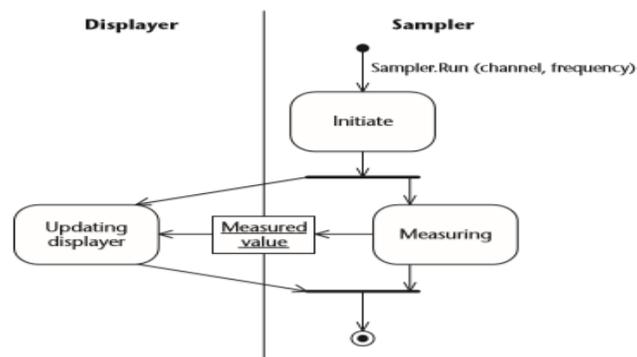
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. [7]



Gambar 2. 2 Usecase Diagram

3. Activity Diagram

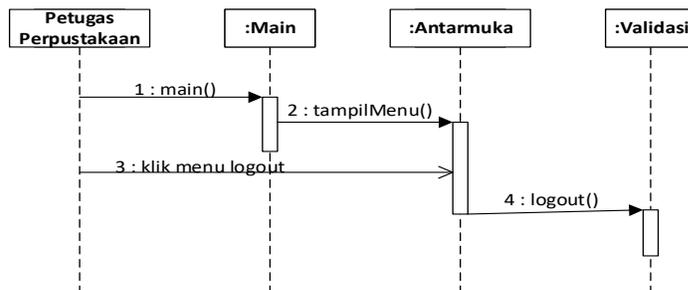
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. [7]



Gambar 2. 3 Contoh Activity Diagram

4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence* diagram maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence* diagram juga dibutuhkan untuk melihat skenario yang ada pada *use case*. [7]



Gambar 2. 4 Contoh Sequence Diagram