www.itk.ac.id

BAB 2
TINJAUAN PUSTAKA

Pada pengantar bab ini merupakan deskripsi singkat dari isi bab 2 Kajian Pustaka dan Dasar Teori. Isi bab 2 Pendahuluan meliputi : Pustaka/Teori 1, Bab ini merupakan penjelasan dari tinjau pustaka yang bersumber dari buku, jurnal maupun artikel yang berkaitan dengan penelitian. Tujuannya untuk memberikan pemahaman atas tahapan-tahapan yang digunakan untuk memberikan solusi dari permasalahan yang ada.

2.1 Digital Koran *atau E-paper*

Electronic paper (*E-paper*) adalah digitalisasi kertas dalam bentuk dokumen seperti PDF, image dan lain-lain. *E-paper* merupakan bentuk inovasi pelayanan kepada masyarakat yang menggunakan internet. Media cetak seperti koran menggunakan *e-paper* sebagai saran publikasi untuk menjangkau *user* yang lebih luas. *E-paper* dianggap lebih praktis karena dapat diakses dimana saja. *User* dapat merasakan tampilan koran cetak secara online tanpa harus membelinya. Hal tersebut dapat meningkatkan pembaca koran. Pembaca yang banyak dapat meningkatkan omzet iklan. Sehingga dapat meningkatkan finansial perusahaan. Hadirnya format *e-paper* membuat bentuk koran menjadi lebih praktis. *E-paper* sendiri juga dilengkapi dengan fitur-fitur seperti print, thumbnail, dan arsip berita (Hershey, 2010).

E-paper Kaltim Post merupakan produk digital dari Kaltim Post edisi koran yang diterbitkan oleh PT Duta Manuntung. *E-paper* Kaltim Post saat dapat diakses melalui *website epaper.kaltimpost.co.id*. Layanan *e-paper* Kaltim Post dapat dimanfaatkan oleh anggota yang telah mendaftarkan diri di epaper.kaltimpost.co.id. Pengguna yang telah terdaftar akan masuk ke dalam *database* sesuai dengan masa aktif yang berlaku. Ketika pengguna melakukan registrasi atau mendaftar, kemudian muncul form pendafataran. Pengguna wajib mengisi form tersebut sebelum menikmati fitur epaper selanjutnya. Setelah form sudah diisi, lalu tekan *button* daftar. Setelah berhasil mendaftarkan akun, pengguna

www.itk.ac.id

www.itk.ac.id

diarahkan untuk mengisi *username* dan password lagi. Setelah berhasil akan muncul form transaksi berlangganan *e-paper* Kaltim Post. Pengguna harus melakukan order langanan terlebih dahulu untuk menikmati layanan *e-paper*. Pengguna bisa juga menghubung *hotline* pemasaran Kaltim Post untuk berlangganan. Pembayaran dilakukan melalui *automatic teller machine* (ATM) atau transfer tunai ke bank (Kaltim Post, 2008).

Mobile application atau yang biasa disingkat *mobile apps* merupakan perangkat lunak yang secara khusus dibuat hanya untuk dibuka melalui tablet maupun *smartphone*. Developer *mobile apps* menggunakan *intergrated development environments* (IDE) dan *software development kit* (SDK) dalam pengembangan *mobile apps*. Aplikasi yang terdapat di *mobile* biasanya diunduh terlebih dahulu melalui store, seperti apple *apps* store, Samsung *apps*, amazon kindle fire, windows store dan google playstore (Irwansyah & Moniaga, 2014).

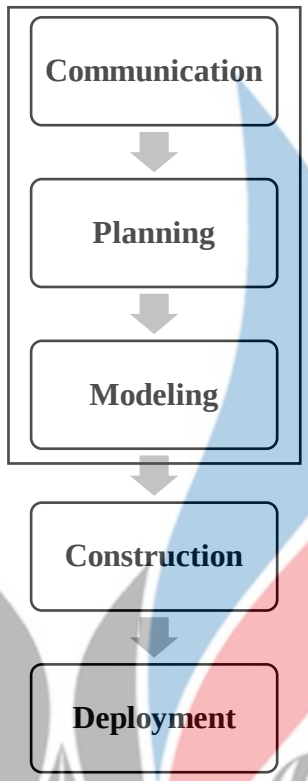
Aplikasi *mobile* adalah aplikasi yang dapat digunakan dimana saja. Aplikasi *mobile* sering digunakan untuk game, web browser, music, radio, berita dan sebagainya. Pengembangan aplikasi *mobile* selanjutnya menggunakan J2ME (*Java 2 Micro Edition*) yaitu program yang menggunakan bahasa pemrograman Java yang terdiri dari *Java Virtual Machine* (JVM). JVM sering dipakai untuk menjalankan program Java pada emulator dan Java API (*Application Programming Interface*) serta tools lain yang digunakan untuk mengembangkan aplikasi java seperti Java emulator, dan Motorolla (Wahana Komputer, 2014).

2.2 Metode *Waterfall*

Metode *waterfall* diciptakan oleh William Royce pada tahun 1970. Secara harfiah, metode *waterfall* dapat diartikan model yang memiliki proses yang sistemasi dengan bentuk linear. Praktiknya, *waterfall* model menjelaskan bahwa setiap proses harus dilaksanakan secara tuntas sebelum melakukan proses selanjutnya Menurut Pressman, model *waterfall* terdiri dari *communication*, *planning*, *modeling*, *construction*, dan *deployment* (Wicaksono, 2017).

www.itk.ac.id

www.itk.ac.id



Gambar 2.1 Metode *Waterfall*

Gambar 2.1 merupakan langkah-langkah pada tahap metode *waterfall*. Penelitian ini berdasarkan gambar 2.1 hanya sebatas langkah komunikasi, perencanaan dan pemodelan. *Communication* merupakan langkah untuk mengetahui permasalahan yang terjadi serta mengetahui lebih mendalam bagaimana permasalahan dapat diselesaikan. Tahap komunikasi ini menjelaskan tahapan diperlukan analisis untuk mengetahui kebutuhan pengguna. Setelah proses komunikasi dilakukan, maka dilanjutkan proses *planning* yaitu tahapan menjelaskan semua kebutuhan *user* dalam bentuk spesifikasi kebutuhan dengan menjelaskan kebutuhan sistem, desain sistem serta arsitektur sistem. Tahap perencanaan ini menghasilkan sebuah dokumen yang disebut dokumen *Software Requirement Specification* (SRS). Dokumen ini menjelaskan kebutuhan sistem berdasarkan fungsional dan non fungsional. Tahapan berikutnya merupakan *modeling* yang berisi sebuah perancangan aplikasi dengan bentuk desain sistem dan desain *interface*. Tahap selanjutnya yaitu *construction*, tahap ini merupakan proses membuat code. Deploymenet, merupakan tahapan final dengan melakukan pemelihataan secara berkala (Pressman, 2015)

www.itk.ac.id

www.itk.ac.id

Communication merupakan suatu proses memilih, menyortir, mengirim suatu informasi sehingga membuat respon komunikator. Langkah ini merupakan proses pengumpulan data dengan melakukan wawancara, observasi langsung, dan studi literatur berupa jurnal, buku dan artikel terkait mengenai permasalahan yang terjadi. Wawancara dapat dilakukan dengan karyawan perusahaan terkait dengan mengajukan beberapa pertanyaan. Observasi langsung juga dapat dilakukan dengan melihat data-data yang terdapat di perusahaan seperti data pengunjung, data pengguna dan lainnya. Studi literatur perlu dilakukan agar tahapan yang akan dilakukan dalam penelitian sesuai dengan penelitian sebelum dan ada sumber ilmiah yang menjelaskannya (Pressman, 2015). Penelitian ini menggunakan metode Kano untuk mengumpulkan data-data kebutuhan *user* di dalam *e-paper* Kaltim Post berbasis *mobile*. Ketika semua kebutuhan *user* sudah dijelaskan, maka selanjutnya merupakan proses *planning*. Proses *planning* adalah tahapan perencanaan suatu sistem dalam bentuk dokumentasi. Dokumentasi ini akan berbentuk dokumen yang berisi secara khusus kebutuhan *user* di dalam sebuah *software*. Dokumen ini harus jelas agar memudahkan proses selanjutnya. Proses ini memiliki *output* berupakan dokumen SRS. Pada tahapan *modeling* berisi desain sistem dan desain aplikasi. Proses berisi alur proses, data, aliran proses dan pemenuhan kebutuhan sesuai dengan hasil dokumen SRS. Tahapan ini berfokus pada perancangan struktur data, arsitektur software serta tampilan desain sistem dan algoritma program. Tugas dari tahap *Modeling* ini yaitu memberikan pemahaman secara jelas mengenai hal-hal yang dikerjakan di dalam sistem. Tahapan ini memiliki output yaitu *Software design description* (SDD). Kelebihan dari metode *waterfall* yaitu memungkinkan untuk mengkategorikan fitur yang dapat dikontrol karena fase pengerjaan proses yaitu satu per satu sehingga meminimalkan kesalahan yang terjadi. Pengembangan bergerak dari konsep mulai dari desain, implementasi, pengujian, instalasi, penyelesaian masalah dan berakhir dengan pemeliharaan. Rangkain dari metode *waterfall* sangat sistematis. Tugas perancangan dapat dilakukan oleh masing-masing anggota yang terdiri dari analis, desainer, programmer, tester atau pemasaran. Setiap tim akan bekerja sesuai dengan tahapan pada metode ini. Sehingga kesalahan bersifat teknis dapat dihindari sedini mungkin (Pressman, 2015).

www.itk.ac.id

www.itk.ac.id

2.3 Model Kano

Metode Kano ditemukan oleh professor Noriaki Kano dari Tokyo Rika University. Metode Kano merupakan metode yang digunakan untuk mendapatkan fitur-fitur prioritas yang akan menjadi fokusan perusahaan pada tahap implementasi awal. Langkah awal adalah melakukan listing terhadap kebutuhan-kebetuhan yang mungkin akan dimasukkan ke dalam aplikasi. Kemudian, menemui calon *user* yang berpontensial menjadi *user*. Lalu, memberikan list kebutuhan fitur yang akan dimasukkan ke dalam aplikasi. Calon *user* akan memilih fitur-fitur apa saja yang sesuai dengan keinginan mereka. Metode Kano membagi persyaratan produk dalam tiga tipe yaitu *must be*, one-dimensional dan *attractive*. *Must be* merupakan kriteria dasar dari suatu produk. Mengetahui faktor must-be ini *user* hanya mencapai pernyataan "Tidak mengecewakan". Bahkan jika faktor ini Tidak ada maka *user* sangat kecewa. *One dimensional* merupakan persyaratan kebutuhan pelanggan berada di tingkatan pemenuhan yang bersifat seimbang. Maksudnya, semakin tinggi parameter pemenuhan kebutuhannya, tentu kebutuhan pelanggan meningkat. Begitu juga sebaliknya. *Attractive* merupakan pemenuhan persyaratan *user* lebih dari kebutuhan yang proporsional. Namun, jika hal tersebut Tidak ada, Tidak membuat *user* kecewa (Nurhayati, 2014).

Tabel 2.1 Model Kano

Kebutuhan Pelanggan		**Pertanyaan Disfungsional (Negatif)**				
		Suka	Harus	Netral	Boleh	Tidak Suka
Pertanyaan fungsional (positif)	Suka	Q	A	A	A	O
	Harus	R	I	I	I	M
	Netral	R	I	I	I	M
	Boleh	R	I	I	I	M
	Tidak suka	R	R	R	R	Q

*) Nurhayari, 2014

Dari tabel 2.1 menjelaskan bahwa A menandakan *attractive* (menarik), M berarti must-be (harus ada), O berarti one-dimensional (satu dimensi), R berarti *reverse* (kebalikan), Q berarti *questionable* (diragukan), I berarti indifferent (biasa saja). Setiap pertanyaan akan dinyatakan dua kali kepada responden. Pertanyaan

www.itk.ac.id

www.itk.ac.id

akan mengandung unsur positif dan negatif. Hasil dari kueisioner ini diambil dari mayoritas jawaban yang dipilih. Keuntungan menggunakan metode Kano yaitu prioritas pada pengembang produk, sehingga perusahaan tidak fokus dalam menginvetasikan fitur yang harus ada namun fokus pada *one-dimensional* atau *attractive* untuk melakukan pengembangan. Kategori fitur *one dimensional* atau *attractive* fokus pada kualitas produk dan mengetahui kebutuhan pelanggan. Keuntungan yang lain yaitu mengetahui syarat produk lebih dipahami karena faktor-faktor yang menyebabkan kebutuhan pelanggan dapat diidentifikasi secara jelas. Pengelompokkannya pun sangat spesifikasi sehingga mudah untuk diimplementasikan dalam bentuk perancangan. Keuntungan mengguna metode Kano yaitu mengoptimalkan kebutuhan pelanggan dengan penyebaran fungsi kualitas (Nurhayi et al., 2014).

2.4 Model Nielsen

Usability merupakan suatu standar untuk menilai kualitas *interface* (tampilan) dari sistem informasi yang digunakan oleh *user*. Terdapat empat kategori yaitu *learnability*, *efficiency*, *memorability*, *error*, dan *satisfaction*. *Learnability* adalah kategori yang menjelaskan kemudahan *user* dalam memahami bagaimana fungsi pada *website* bekerja. Indikatornya yaitu *Easy to understand*, *Easy to look for specific information* dan *easy to identify navigational mechanism*. *Easy to understand* artinya bagaimana *user* dapat mengerti tujuan maupun informasi yang terdapat di *website* tersebut dengan mudah. Sedangkan *Easy to look for specific information* merupakan indikantor yang menjelaskan bagaimana *user* dapat memperoleh informasi yang disajikakn oleh konten-konten yang terdapat dalam *website* dengan mudah serta mendapatkan manfaatnya. Indikator *easy to identify navigational mechanism* menjelaskan *user* dapat mengoperasikan langkah-langkah navigasi pada setiap fitur yang tersedia di dalam *website* dengan mudah.

Efficiency adalah kategori yang menjelaskan pengukuran berdasarkan kecepatan dan ketepatan pengguna dalam mengakses sistem. Indikator aspek *efficiency* yaitu *easy to reach quickly* dan *easy to navigate*. *Easy to reach quickly* yaitu *user* mendapatkan informasi serta fitur yang dibutuhkan secara cepat. *Easy to navigate* menjelaskan bahwa wawasan *user* terhadap *website* dalam menjelajahi

www.itk.ac.id

www.itk.ac.id

fitur dan konten yang disuguhkan pada *website* dengan mudah. *Memorability* adalah kategori yang menjelaskan tingkat ingatan pengguna dalam mengoperasikan sistem seperti tata letak fitur dan tampilan sistem. Indikator keberhasilan aspek *memorability* yaitu *easy to remember* dan *easy to reestabliash*. *Easy to remember* yaitu bagaimana *user* dapat dengan mudah diingat oleh *user* perihal fitur serta konten yang terdapat dalam *website*. *Easy to reestablish* perihal *website* dapat diakses kembali oleh *user* dengan tampilan *website* yang sama saat *user* mengakses *website* tersebut. *Error* adalah kategori yang berkaitan tentang pengetahuan *user* tentang kerusakan yang terjadi dalam *website* serta upaya perbaikan yang dapat dilakukan *user*. Indikator keberhasilan aspek *error* ini yaitu *few number of error*s detected dan *easy to fix*. *Satisfaction* adalah kategori pengukuran tingkat kebutuhan *user* terhadap desain yang ada di dalam *website*. Indikator keberhasilan aspek *satisfaction* yaitu s*ystem pleasant to use* dan c*omfort to use* (Nielsen, 1993).

2.5 *Software Requirement Specification* (SRS)

Dokumen ini berisi proses untuk mempelajari keinginan pengguna yang berawal dari defines dari sistem, hardware dan kebutuhan software. SRS merupakan dokumen yang berisi dokumentasi dari kebutuhan pokok seperti fungsi, kinerja dan atribut yang terdapat dalam sistem. Dokumen SRS juga berisi mengenai kendala yang dihadapi di dalam sistem. SRS yang baik harus memiliki delapan syarat yaitu tepat, tidak rancu, lengkap, tetap, sistematis, jelas kebenarannya, dapat dimodifikasi dan dapat dilacak. SRS merupakan hasil akhir dari analisis kebutuhan yang sudah dideskripsikan dengan jelas (Wicaksono, 2017). Dokumen SRS ini menjelaskan semua interaksi yang dilakukan pengguna dengan perangkat lunak. Selain pengguna, SRS juga berisi kebutuhan fungsional dan non fungsional dari sistem. Kebutuhan fungsional harus dipenuhi untuk menentukan cara kerja dari sistem. Kebutuhan non fungsional berisi batasan sistem saat mengimplementasi perangkat lunak yang terdiri dari kebutuhan kinerja, standar kualitas dan batasan desain. Dokumen SRS berisi tujuan, cakupan sistem, gambaran umum sistem, referensi, definisi, *use case*, kebutuhan fungsional dan kebutuhan nonfungsional (Stellman & Greene, 2005).

www.itk.ac.id

www.itk.ac.id

2.6 *Software design description* (SDD)

Tujuan dari desain perangkat lunak adalah membangun model yang memenuhi semua kebutuhan pelanggan dengan tujuan implementasi yang sesuai. Sistem perangkat lunak berkembang dalam skala, kompleksitas, dan distribusi desian yang tepat menjadi sangat penting dalam merancang perangkat lunak. Perangkat lunak jenis apapun dan domain aplikasi apapun harus memiliki desain arsitektur keseluruhan yang menjadi acuan pembangunan dan pengembang perangkat lunak. Keberhasilan perangkat lunak sangat bergantung pada keberhasilan desain arsitekturnya. Desain arsitektur merupakan hubungan antar element struktural perangkat lunak, gaya dan pola desain yang dapat digunakan untuk mencapai kebutuhan untuk pembuatan sistem serta batasan yang mempengaruhi cara arsitektur dapat diimplemntasikan. Representasi keseluruhan dari perangkat lunak yang akan di bangun yaitu *Software design descriptions*(SDD) berdasarkan standar dari *Institute of Electrical and Electronics Engineers* (IEEE) 1016. SDD berperan sebagai *blueprint* pada tahap implementasi. Berdasarkan IEEE 1016, SDD berisi tinjauan desain, tujuan, ruang lingkup, deskripsi dekomposisi (modul, data proses), deskripsi koneksi, atribut, deskripsi antamuka pengguna dan penjelasan desain secara detail (Qian, Fu, Tao, Xu, dkk, 2010)

Dokumen SDD umumnya berisi mengenai deskripsi mengenai produk software dengan panduan yang jelas. Dokumen mendeskripsikan proses-proses desain dan proses pembuatan aplikasi secara terperinci Dokumen ini berisi desain sistem dan desain aplikasi Desain sistem digambarkan dengan menggunakan bahasa pemodelan Unified. Model Language (UML) yang terdapat *use case diagram*, *activity diagram, class diagram* dan *sequence diagram*. Desain aplikasi dibuat dalam bentuk prototipe menggunakan AdobeXD (Pressman, 2015).

2.7 *Unified Model Language* (UML)

Pemodelan adalah langkah sederhana dari suatu permasalahan. Tujuan dari pemodelan yaitu sarana komunikasi secara visual dengan berbagai pihak yang terlibat dalam pengembangan sisem tersebut. Pemodelan juga digunakan sebagai alat dokumentasi untuk menjelaskan lebih jauh mengenai suatu sistem secara jelas. Diperlukan juga untuk keperluan testing sistem yang telah dilakukan

www.itk.ac.id

www.itk.ac.id

pengembangan. Dalam melakukan pengembangan sistem dibutuhkan *tool* untuk menjelaskan sistem tersebut. Paling sering digunakan adalah notas-notasi *Unified Modeling Language* (UML). UML merupakan bahasa pemodelan visual yang menjelaskan secara spesifik dokumentasi rancangan dari suatu sistem perangkat lunak. UML sangat fleksibel digunakan dalam pengembangan sistem karena dapat menyesuaikan dengan berbagai domain aplikasi. Konsep dalam UML terdapat semantil, notasi dan paduan masing-masing diagram. UML memiliki dua macam diagram utama yaitu *structure* diagram dan *behavior* diagram (Tohari, 2014).

Structure diagram merupakan diagram yang memiliki susunan statis dari sistem dan bagian dari abstraksi implementasi yang berlainan dan cara bagian tersebut saling berinteraksi. *Structure* diagram Tidak menggunakan konsep hubungan waktu dan tingkah laku yang berubah. Tetapi, menunjukan relasi kejadian dari *classifiers* yang ditunjukkan dalam *structure* diagram. *Structure* diagram terdapat *class diagram, object* diagram, *package* diagram, model diagram, *composite structure, component diagram, deployment* diagram, dan *profle* diagram. *Behavior* diagram yaitu diagram yang menjelaskan tingkah laku dinamis dari objek-objek yang terdapat di sistem yang terjadi sepanjang waktu. Jenis-jenis behavior diagram di antaranya *Use case Diagram, Activity diagram, state machine* diagram, dan *interaction* diagram (Akil, 2018).

2.8 *Use Case* Diagram

Use case merupakan alur yang dikerjakan oleh sistem sesuai dengan keputusan dari *user*. Tahap awal dalam membuat diagram *use case* adalah menjelaskan aktor dan prosesnya. Langkah membuat diagram *use case* yang pertama yaitu menjelaskan peran aktor. Kemudian peran aktor tersebut harus dijelaskan agar mencapai beberapa fungsi dari sistem tersebut (Triandini, 2012). Sebuah unit eksternal dari sistem (berupa *interface*) yang akan merespon perintah dari *aktor* berupa sebuah event. *Use case* dapat menyampaikan pesan yang diberikan oleh objek. Secara dinamis *use case* dapat dijelaskan secara terperinci oleh *Sequence Diagram*, *Activity Diagram*, *state machine* diagram dan *Communication* diagram serta deskripsi berbasis teks (Akil, 2018).

www.itk.ac.id

www.itk.ac.id

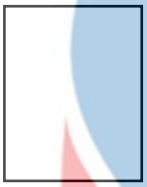
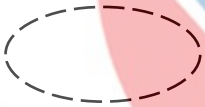
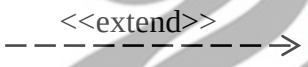
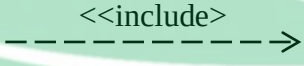
Diagram ini menotasikan aktor sebagai *user* dan proses pengguna yang disebut deskripsi *use case*. Arah panah menunjukan bahwa aktor memulai komunikasi hingga mencapaikan tujuan sistem yang dapat dilakukan aktor. Termasuk dalam diagram *use case* di antaranya diagram secara singkat menjelaskan semua *use case* dan aktor, secara singkat diagram menunjukkan aktor yang terkait secara konseptual. Diagram berperspektif aktor yang menunjukkan semua *use case* yang melibatkan aktor. Diagram menjelaskan *case* penggunaan menunjukkan serangkaian kasus pengguna yang terkait dengan sistem. Diagram *use case* yang menggambarkan bagaimana hubungan antara aktor dan *case* pengguna lainnya (Bittner, 2003).

Elemen-elemen diagram *use case* yaitu sistem, aktor, *use case*, *association*, *stereotape*, *dependency (include)*, dan *generalization*. *Use case* dapat berkaitan dengan kasus lainnya karena adanya hubungan. Ada berbagai macam jenis hubungan di dalam *use case*. Di antaranya *association* yang berfungsi menjalin komuniasi antar seorang aktor dengan satu *use case* dimana aktor tersebut berpartisipasi. Hubungan eksternal berfungsi menyisipkan sebuah fungsionalitas tambahan ke dalam sebuah *use case* yang bersifat pilihan. Hubungan *include* menyampaikan sebagai hubungan tambahan ke dalam sebuah *use case* dasar saat fungsionalitas tersebut bersifat wajib. Hubungan *use case generalization* merupakan hubungan antar *use case* yang lebih spesifik saat *use case* yang lebih spesifik mewarisi *use case* umum tersebut. Stereotape berfungsi sebagai kualifier dari suatu elemen model. Dalam memodelkan konteks dari sistem perlu mengindentifikasi batasan dari sistem dengan menjelaskan bagian tingkah laku yang berada di dalam sistem dan di luar sistem. Kemudian identifikasi aktor-aktor yang berkaitan dengan sistem tersebut. Jelaskan aktor sesuai dengan grup yang membutuhkan bantuan dari sistem untuk menyelesaikan tugasnya. Kemudian sesuaikan juga dengan grup yang dapat berinteraksi dengan *hardware* dan jelaskan juga fungsi yang berfungsi sebagai administrasi dan *maintenance*. Setelah itu, kelompokkan aktor yang sama sesuai dengan tingkatan generalisasi. Diagram *use case* berfungsi untuk menyusun requirement sebuh sistem, mengkomunikasikan rancangan dengan klien serta merancang *test case* untuk fitur yang ada dalam sistem (Tohari, 2014).

www.itk.ac.id

www.itk.ac.id

Tabel 2.2 Simbol UML

Gambar	Nama	Keterangan
	*Use case*	Deskripsi dari urutan interaksi yang ditampilkan sistem untuk menghasilkan suatu hasil yang terukur bagi aktor.
	Sistem	Menjelaskan secara khusus paket yang menampilkan sistem secara terbatas.
	*Collaboration*	Interaksi antar aturan dan elemen lain yang bereaksi sama untuk menghasilkan sinergi.
	Aktor	Menunjukan pengguna yang menggunakan serta berinteraksi dengan *use case*
	*Extend*	Penambahan sebuah fungsionalitas tambahan ke dalam sebuah *use case* dasar yang memiliki sifat opsional.
	Association	Penghubung komunikasi antar seorang aktor dengan satu *use case* dimana aktor tersebut berperan.
	*Include*	Penambahan sebuah fungsionalitas ke dalam *use case* dasar yang memiliki sifat wajib.
	*Use case generalization*	Interaksi antar sebuah *use case* umum dengan *use case* yang

www.itk.ac.id

www.itk.ac.id

Gambar	Nama	Keterangan
		lebih spesifik saat *use case* yang lebih spesifik mewarisi *use case* umum.

*) Tohari, 2014

Tabel 2.2 menjelaskan simbol-simbol yang terdapat pada UML yang terdiri dari *use case* yang berfungsi mendeskripsi dari urutan interaksi yang ditampilkan sistem untuk menghasilkan suatu hasil yang terukur bagi aktor. Kemudian ada sistem menjelaskan secara khusus paket yang menampilkan sistem secara terbatas. Collaboration mendeskripsikan interaksi antar aturan dan elemen lain yang bereaksi sama untuk menghasilkan sinergi. Aktor mendekripsikan menunjukan pengguna yang menggunakan serta berinteraksi dengan *use case*. *Extend* mendeskripsikan penambahan sebuah fungsionalitas tambahan ke dalam sebuah *use case* dasar yang memiliki sifat opsional. *Association* mendeskripsikan penghubung komunikasi antar seorang aktor dengan satu *use case* dimana aktor tersebut berperan. Include mendekripsikan penambahan sebuah fungsionalitas ke dalam *use case* dasar yang memiliki sifat wajib. *Use case* generalization mendeskripsikan interaksi antar sebuah *use case* umum dengan *use case* yang lebih spesifik saat *use case* yang lebih spesifik mewarisi *use case* umum.

2.9 *Activity Diagram*

Menggambarkan *workflow* (alur kerja) dari sebuah sistem disebut *Activity diagram*. Diagram ini menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* juga dapat menjelaskan rangkaian proses setiap aktivitas yang menggambarkan proses yang terjadi di sistem. Diagram ini akan mengelompokkan tampilan dari sistem *interface* saat semua aktivitas mempunyai rancangan *interface* sendiri. Setiap rancangan memerlukan sebuah pengujian yang diperlukan untuk mendefiniskan kasus yang di-*testing*. *Activity diagram* menjelaskan berbagai aliran aktivitas, menjelaskan masing-masing aliran mulai dan berakhir, keputusan yang mungkin terjadi, dan cara mengakhirinya. *Activity diagram* berisi *state* yang merupakan aktivitas (Sugiarti, 2018).

www.itk.ac.id

www.itk.ac.id

Elemen-elemen yang terdapat pada diagram ini yaitu status mulai dan berakhir, kegiatan mendeskripsikan sebuah langkah dan *workflow*, *transition* menjelaskan perubahan yang terjadi, keputusan ditunjukkan dengan *alternatif* dalam *workflow*, *synchronization bars* dan *swimlanes*. Synchronization bars merupakan sub aliran parallel yang digunakan untuk menjelaskan concurrent threads pada alur kerja proses bisnis. Sedangkan swimlanes merupakan gambaran tugas bisnis yang bertanggung jawab pada aktivitas yang berjalan (Tohari, 2014).

Tabel 2.3 Simbol *Activity diagram*

Gambar	Nama	Keterangan
	*Notasi state*	Menggambarkan interaksi yang dilakukan dalam aliran kerja.
	Transisi	Menghubungkan interaksi antar interaksi yang lain yang saling berkaitan.
	Notasi *decision*	Menjelaskan sebuah keputusan yang dibuat dalam aliran kerja.
	Notasi *start state*	Menggambarkan aliran kerja dimulai.
	Notasi *end state*	Menggambarkan aliran kerja selesai.

*) Tohari, 2014

Tabel 2.3 menjelaskan notasi-notasi yang terdapat pada *activity diagram*. Notasi *state* menggambarkan interaksi yang dilakukan dalam aliran kerja. Transisi menghubungkan interaksi antar interaksi yang lain yang saling berkaitan. Notasi *decision* menjelaskan sebuah keputusan yang dibuat dalam aliran kerja. Notasi *start state* menggambarkan aliran kerja dimulai. Notasi *end state* menggambarkan aliran kerja selesai.

www.itk.ac.id

www.itk.ac.id

2.10 Class Diagram

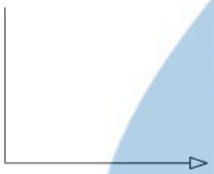
Class diagram menggambarkan struktur statis dan hubungannya yang ada di dalam sistem. Satu *class diagram* menjelaskan *interface* yang terjadi setiap hubungan yang terjadi. Dalam pemodelan statis, *class diagram* digunakan untuk rekapan dari sistem, kolaborasi dan skema basis data *logical*. *Class diagram* memiliki tiga area pokok yaitu nama, atribut dan metode. Atribut dapat memiliki salah satu sifat yaitu *private*, *protected* dan *public*. *Private* merupakan tindakan dimana sebuah kelas Tidak dapat memanggil kelas yang berada di luar kelas. Protected hanya dapat dipanggil oleh sub bagian yang memiliki sifat yang berkaitan. *public* merupakan sifat yang dapat dipanggil oleh kelas mana saja. Mendefiniskan *class diagram* dapat diawali dengan melihat *use case diagram*. Di dalam *class diagram* terdapat objek, setiap objek ini memiliki metode atau operasinya sendiri. (Tohari, 2014).

Tabel 2.4 Simbol *Class diagram*

Gambar	Nama	Keterangan
	*Package*	Gambar yang menjelaskan sebuah sampul dari suatu atau lebih kelas
	*Class*	Kelas yang terdapat pada struktur sistem
	Nama *interface*	Memiliki pengertian yang sama dengan konsep *interface* dalam pemrograman berorientasi objek.
	Association	Menjelaskan relasi antar kelas dengan makna umum.
	Directed Association	Menjelaskan relasi antar kelas dengan makna kelas yang satu

www.itk.ac.id

www.itk.ac.id

		digunakan oleh kelas yang lainnya.
	*Inheritance*	Menjelaskan relasi antar kelas dengan makna umum dan khusus.
	*Aggregation*	Menjelaskan relasi antar kelas dengan makna semua bagian.

*) Tohari, 2014

Tabel 2.4 menjelaskan simbol-simbol yang terdapat pada *class diagram*. *Package* menjelaskan gambar yang menjelaskan sebuah sampul dari suatu atau lebih kelas. *Class* menjelaskan kelas yang terdapat pada struktur sistem. Nama *interface* mendeskripsikan memiliki pengertian yang sama dengan konsep *interface* dalam pemrograman berorientasi objek. *Association* menjelaskan relasi antar kelas dengan makna umum. *Directed association* menjelaskan relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lainnya. *Inheritance* menjelaskan relasi antar kelas dengan makna umum dan khusus. *Aggregation* menjelaskan relasi antar kelas dengan makna semua bagian.

2.11 Sequence Diagram

Sequence diagram sering digunakan sebagai sarana mendokumentasikan perilaku dalam *use case*. Dalam diagram tersebut terdapat urutan pengunaan sistem yang disebut *scenario* diagram. Hal tersebut digunakan sebagai narasi dari sistem. *Sequence diagram* memiliki dua dimensi yaitu dimensi vertikal dan horizontal. Dimensi vertikal menyatakan sumbu waktu, yang berjalan dari atas ke bawah diagram. Dimensi horizontal menyatakan peran *classifier* yang mewakili bojek yang saling berkolaborasi. Singkatnya, diagram ini menunjukkan interaksi di antara objek yang berkaitan antara dua titik yang menyatakan waktu. Oleh karena itu, memahami *Sequence diagram* harus mempertimbangkan perilaku pengunaan.

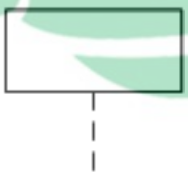
www.itk.ac.id

www.itk.ac.id

Membaca *Sequence diagram* dapat memberikan informasi terperinci antara aktor dan sistem atau antara objek yang berkesinambungan, dalam kurun waktu tertentu (Unhelkar, 2005).

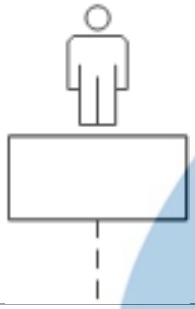
Sequence diagram mendeskripsikan *behavior* objek pada *use case* dengan menjelaskan waktu hidup dan pesan yang disampaikan antar objek. Banyaknya diagram sekuens yang didekripsikan yaitu sebanyak pendefinisian *use case* yang memiliki proses tunggal. Semakin banyak *use case diagram* yang dibuat maka semakin banyak juga diagram sekuens yang harus dibuat. Satu objek hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Deskripsi antar objeknya dijelaskan dengan anak panah dari objek yang mengirimkan pesan ke objek yang menerima pesan (Tohari, 2014). Diagram ini hanya menjelaskan peranan satu aktor dengan berbagai pesan interaksi yang akan dilakukan oleh aktor tersebut terhadap sistem. Simbol-simbol yang ada di *Sequence diagram* di antaranya *entity class, boundary class, control class, massage, recrusive, activation,* dan *lifelline*. *Entity class* adalah bentukan dari sistem yang isinya gabungan kelas yang menunjukkan entitas-entitas yang menjadi awal dari sistem. *Boundary class* yaitu gabungan dari kelas yang merupakan *interface* satu atau lebih aktor dengan sistem. *Control class* adalah objek yang berisi aturan yang Tidak berhubungan langsung dengan sistem misalnya kebijakan perusahaan maupun peraturan pemerintah. Massage merupakan simbol pengiriman pesan antar *class*. *Recursive* adalah representasi pengiriman pesan yang dikirim untuk objek itu sendiri. Activation berperan mensubtitusikan sebuah hasil operasi dari objek. Lifeline adalah garis titik-titik yang menghubungkan antara objek sepanjang waktu terhadap aktivasi sistem. *Sequence diagram* digunakan sebagai penunjuk setiap kemungkinan proses interaksi yang terjadi di dalam sistem atau menjelaskan setiap alur interaksi yang terjadi di sistem (Urva, 2015).

Tabel 2.5 Simbol *Sequence diagram*

Gambar	Nama	Keterangan
	*Object lifeline*	Menunjukan gambar interaksi yang disertai nama objek, nama kelas, atau keduanya.

www.itk.ac.id

www.itk.ac.id

	*Aktor lifeline*	Menggambar aktor yang berada di luar lingkup sistem serta aktor tentunya berhubungan dengan sistem yang dibangun.
	Message	Menjelaskan suatu objek atau kelas dapat meminta objek lainnya untuk menjalankan fungsi spesifik.
	*Activation*	Menggambarkan objek dalam keadaan aktif dan berinteraksi pesan.

*) Urva, 2015

Tabel 2.5 merupakan simbol dari *sequence diagram*. *Object lifeline* menunjukan gambar interaksi yang disertai nama objek, nama kelas, atau keduanya. *Aktor lifeline* menggambar aktor yang berada di luar lingkup sistem serta aktor tentunya berhubungan dengan sistem yang dibangun. *Message* menjelaskan suatu objek atau kelas dapat meminta objek lainnya untuk menjalankan fungsi spesifik. *Activation* menggambarkan objek dalam keadaan aktif dan berinteraksi pesan.

2.12 *Entity Relationship Diagram* (ERD)

Berdasarkan Brady dan Loonam (2010), *entity relationship diagram* (ERD) yaitu teknik yang digunakan untuk memodelkan kebutuhkan data dari suatu kelompok, biasanya dalam tahap analisis sebagai kebutuhan proyek pengembangan perangkat lunak. Skema pemodelan merupakan metode yang digunakan membuat model atau menggambarkan *database*. Komunikasi yang dilakukan biasanya berupa diagram grafik agar mudah dipahami. Salah satunya *entity relationship diagram* (ERD) yang merupakan bagian dari model semantic dalam *database*. ERD bukan satu-satunya alat pemodelan semantik, namun ERD merupakan alat pemodelan yang umum digunakan dan sangat populer. ERD adalah alat pemodelan yang digunakan untuk menggambar data secara abstrak. Data yang dideskripskan secara abstark disebut model konseptual. ERD juga dapat digunakan untuk

www.itk.ac.id

www.itk.ac.id

mendokumentasikan *database* yang ada dengan memperkenalkan entitas yang ada di dalam perangkat lunak (Bagui & Earp, 2003).

ERD adalah alat yang digunakan untuk menjelaskan kebutuhan data dan hipotesis yang terdapat di dalam sistem yang akan dirancang secara sistematis. Model data ERD ini juga diatur pada tahap *waterfall*. Pembuatan ERD memerlukan pemahaman terhadap sistem serta atribut penyusunnya. Cara membuat ERD yang pertama yang dilakukan yaitu mendeskripsikan serta menentukan seluruh himpunan entitas yang terlibat. Setelah itu menentukan *primary key* masing-masing entitas. Lalu, menjelaskan seluruh relasi di antara himpunan entitas-entitas yang ada beserta *foreign key*. Selanjutnya mendeskrpsikan relasi garis antar himpunan. Terakhir melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (Supardi, 2008).

2.13 AdobeXD

Adobe XD merupakan aplikasi yang dapat digunakan untuk menggambarkan protipe suatu desain aplikasi. Kelebihan dari aplikasi ini yaitu menyediakan ukur kanvas yang sesuai dengan ukuran perangkat yang diinginkan. Adobe XD terdapat kanvas berbentuk *windows, android, iOS* dan lain sebagainya. Hal tersebut mempermudah dalam melakukan proses prototipe karena Tidak perlu mengatur ukurannya sendiri. Keunggulannya yang lain yaitu di dalam adobe XD terdapat pilihan *user interface* kits untuk windows maupun iOS. Terdapat juga tutorial menggunakan tools di dalam adobe XD. Adobe XD juga memiliki tampilan yang simple sehingga mempermudah *user* untuk menggunakannya. Pada adobe XD terdapat fitur prototyping yang berfungsi menghubungkan setiap halaman bahkan tombol yang dapat diklik seperti tampilan asli sebuah *website* maupun aplikasi. Kelemahan dari adobe XD ini adalah Tidak memiliki fitur animasi dan export CSS. Selain itu adobe XD Tidak dapat melakukan inputan untuk field pada prototipenya (Adhipratama, 2018).

2.14 Uji Validitas dan Reliabilitas

Kegiatan mengguna kuesioner menggunakan validitas dan reliabilitas untuk menunjukan tingkat kebenaran dari hasil perhitungan kuesioner tersebut. Uji

www.itk.ac.id

www.itk.ac.id

validitas merupakan suatu kegiatan mengukur koefisien korelasi antara nilai suatu pertanyaan dalam kuesioner dengan skor total. Metode uji validitas yang digunakan dalam penelitian ini yaitu korelasi pearson atau korelasi product moment. Hasil validitas dapat diketahui pada semua item pertanyaan, jika r tabel < r hitung maka valid. Uji validitas dapat dilakukan dengan menggunakan korelasi produk moen pearson. Analisis produk momen pearson didaptkan dengan menghubungan masing-masing nilai pernyataan kuesioner. Pernyataan dalam kuesioner dapat disimpulkan valid apabila menjabarkan Rhitung lebih besar Rtabel dengan $\alpha = 1\%$.

Reliabilitas merupakan alat ukur yang berkaitan dengan masalah kekeliruan pengukuran. Uji reliabilitas dilakukan setelah perhitungan uji validitas sudah dilakukan. Tujuan dari uji reliabilitas yaitu menjelaskan hasil uji validitas dari pernyataan kuesioner dipastikan konsisten, stabil dan terpercaya. Jika hasil dari uji reliabilitas digunakan secara kontinu tidak mengubah hasil datanya dan data yang didapatkan pun sama. Penelitian ini menggunakan uji reliabilitas dengan menggunakan indikator Cronbach'S Alpha. Cronbach's Alpha memiliki nilai tingkat reliabilitas yang ditunjukkan pada tabel.

Tabel 2.6 Nilai Cronbach's Alpha

Nilai Cronbach's Alpha	Tingkat Reliabilitas
0.0-0.20	Kurang Andal
>0.20-0.40	Agak Andal
>0.40-0.60	Cukup Andal
>0.60-0.80	Andal
>0.80-1.00	Sangat Andal

*) Hair, William & dkk, 2010

Tabel 2.6 mendeskripsikan nilai tingkat keandalan pada Cronchbach's Alpha saat nilai Rhitung pada nilai penelitian memiliki nilai 0.0-0.20 maka memiliki tingkat reliabilitasnya kurang andal, jika nilainya >0.20 sampai 0.40 memiliki tingkat reliabilitas dengan nilai agak andal, jika nilainya >0.40 sampai 0.60 memiliki tingkat reliabilitas dengan nilai andal, jika nilainya >0.80-1.00 memiliki tingkat reliabilitas dengan nilai sangat andal (Hair, William, & dkk, 2010).

www.itk.ac.id

www.itk.ac.id

2.15 Penelitian Terdahulu

Berikut adalah rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan dari laporan serta artikel terkait mengenai perancangan aplikasi berbasis *mobile* dengan menggunakan metode *waterfall* sebagai acuan penelitian ini (lihat tabel 2.7).

Tabel 2.7 Penelitian terdahulu

No	Peneliti	Tahun	Metode	Studi Kasus	Hasil
1.	H. Ilham, E. Soesilo, M. Eng, dkk	2015	Metode *Agile*	MTsN Tandikat	Rancangan sistem informasi perpustakaan digital (*digital library*) berbasis web menggunakan PHP MySQL.
2.	E. G. Saptomi	2016	Metode *Waterfall*	Harian Lampung News Paper	Rancangan aplikasi portal berita harian lampung newspaper pada perangkat bergerak berbasis android
3.	A. D. Haq dan E. Fadilah	2018	Digital Subscription	Penuruan oplah koran pada Harian Kompas	Digital Subscription sebagai langkah Kompas.id untuk menjadi portal media berlangganan.
4.	L. Fadhillah	2018	Metode *Waterfall*	Android studio dengan bahasa pemograman java dan database MySQL	Rancangan portal berita Departemen Matematika Fakultas MIPA Universitas Sumatera Utara berbasis android yang memudahakan mahasiswa untuk mendapat informasi mengenai fakultas tersebut.
5.	D.Susanti & Elmiyati	2020	Metode *Rapid Application*	PT Trita Musi Prasada	Perancangan website media informasi dan pemesanan dengan alat rrancang yaitu flowchart, diagram

www.itk.ac.id

www.itk.ac.id

No	Peneliti	Tahun	Metode	Studi Kasus	Hasil
			Development (RAD)		konteks, *data flow diagram* (DFD), dan *entity relationship diagram* (ERD)
6.	I. G. Pradhana	2020	Metode *Scrum*	Kawasan Pemurbaya Surabaya	Perancangan serta implementasi aplikasi berbasis website mangrove pada Kawasan Pamurbaya Surabaya menggunakan metode *scrum*
7.	R. Yulia, S. R. Natasia, dkk	2021	Metode *Waterfall*	Perancangan e-paper Kaltim Post berbasis mobile menggunakan metode *waterfall*	Perancangan ini menghasilkan hasil akhir berupa dokumen SRS, SDD serta mock-up tampilan e-paper Kaltim Post berbasis mobile.

Tabel 2.7 menjelaskan bahwa terdapat penelitian terdahulu yaitu mengenai perancangan sistem informasi perpustakaan digital MTsN Tandikan menggunakan metode agile dengan metode pengembangan exterem programming dengan tahapan *planning*, *designing*, *coding* dan *testing*. Penelitian ini menggunakan bahasa pemodelan UML serta dalam melakukan perancangan sistem informasi menggunakan XAMPP, notepad++ dan browser (Hafid, Soesilo, & dkk, 2015). Penelitian terdahulu selanjutnya yaitu mengenai digital subscription yang ditulis oleh Alifiyya Dhiya Haq dan Efi Fadilah. Penelitian tersebut menjelaskan langkah Kompas.id ke ranah digital dengan langkah digital subscription. Kompas memberikan paket-paket berlangganan sesuai dengan keinginan pembaca. Langkah ini sebagai ekstensi dari harian Kompas. Pembayaran dan berlangganan tetap ada sebab Kompas ingin memberikan harga untuk aspek jurnalisme serprti pada korannya (Haq & Fadillah, 2018).

Ada penelitian terdahulu mengenai perancangan website media informasi dan pemesanan dengan studi kasus PT Trita Musi Prasada. Penelitian ini menggunakan metode RAD. Metode RAD memiliki tahapan yaitu perencanaan kebutuhan, perancangan dan perbaikan, dan implementasi. Pada tahap perencanaan didapatkan

www.itk.ac.id

www.itk.ac.id

dengan melakukan wawancara, observasi, studi pustaka dan dokumentasi. Tahap perencanaan dijelaskan dengan membuat *flowchart*, *data flow diagram* (DFD), dan *entity relationship diagram* (ERD) (Susanti & Elmiyati, 2020). Ada penelitian mengenai koran digital berbasis android ditulis oleh Erland Gili Saptomi. Penelitian tersebut membuat portal berita berbasis android untuk memudahkan masyarakat dalam mendapatkan informasi secara real time dengan menyajikan berita yang diambil dari *website* lampungnewspaper.citm.co.id dengan metode *waterfall*. Metode ini digunakan karena lebih sistematis dalam membangun suatu aplikasi untuk perusahaan yang memiliki scope yang kecil (Saptomi, 2016).

Penelitian yang ditulis oleh Lilis Fadhillah. Penelitian ini dilakukan karena Departemen Matematika Fakultas MIPA Universitas Sumatera Utara ingin memberikan informasi secara cepat dan mudah kepada mahasiswanya. Oleh sebab itu, penelitian ini membuat portal berita berbasis android (Ayu, 2018). Ada penelitian terdahulu mengenai perancangan dan implementasi pembuatan aplikasi website mangrobe pada Kawasan Pamurbaya Surabaya. Penelitian menggunakan metode *scrum* karena dengan menggunakan metode ini waktu pengerjaan menjadi lebih singkat. Tahapan metode scrum yaitu *product backlog, sprint, sprint planning, daily scrum, sprint review,* dan *sprint restrospective* (Pradhana, 2020). Penelitian dari penulis menjelaskan bahwa perancangan e-paper berbasis mobile ini menggunakan metode *waterfall* serta menggunakan model Kano untuk mengidentifikasi kebutuhan fitur dari pengguna dan perusahaan. Sedangkan untuk mengetahui kebutuhan desain menggunakan model Nielsen. Penelitian ini memiliki hasil berupa dokumen SRS, dokumen SDD serta *mock-up* tampilan (Yulia, Natasia, & dkk, 2021).

www.itk.ac.id