BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Dalam menentukan objek penelitian, penulis melakukannya di Chérie Cake's yang beralamat di Jl. Branjangan II No. 26 Komp. Rajawali Halim PK, Jakarta Timur 13610.

Penelitian yang dilakukan di Chérie Cake's bertujuan untuk membuat suatu aplikasi. Aplikasi yang akan dibuat adalah aplikasi pemesanan dan penjualan berbasis mobile android yang dimaksudkan untuk mempermudah konsumen dalam kesibukannya untuk memesan dan mempercepat pengelolaan data bagi pemilik toko.

3.1.1. Sejarah Singkat Perusahaan

Awalnya, usaha ini dikembangkan dari hobi membuat kue pada acara ulang tahun keluarga dan pada hari-hari raya, yang kemudian mendapat pesanan dari teman dan keluarga dekat pada sekitar akhir tahun 2010. Perkembangan selanjutnya, pada pertengahan tahun 2011, dicobalah untuk lebih menekuni hobinya menjadi pekerjaan sampingan selain menjadi pegawai negeri di salah satu Lembaga Negara.

Sang pemilik, yang memiliki nama lengkap Riri Ardhya Febriani, memulai pencarian alat-alat yang lebih lengkap untuk mempermudah dalam pengolahan kuenya. Dalam proses penjualannya, usaha ini tidak menyediakan *stock*, dengan

alasan, kue tidak bertahan terlalu lama karena tidak menggunakan bahan pengawet, sehingga penjualannya tergantung ada tidaknya pesanan. Jika ada pesanan, maka proses pembuatan kue bisa dilakukan.

3.1.2. Visi dan Misi Perusahaan

Dalam menjalankan kegiatan usahanya, Chérie Cake's memiliki visi dan misi yang tertera di bawah ini.

a. Visi Chérie Cake's

"Memperluas pemasaran dan memelihara konsumen dengan baik."

b. Misi Chérie Cake's

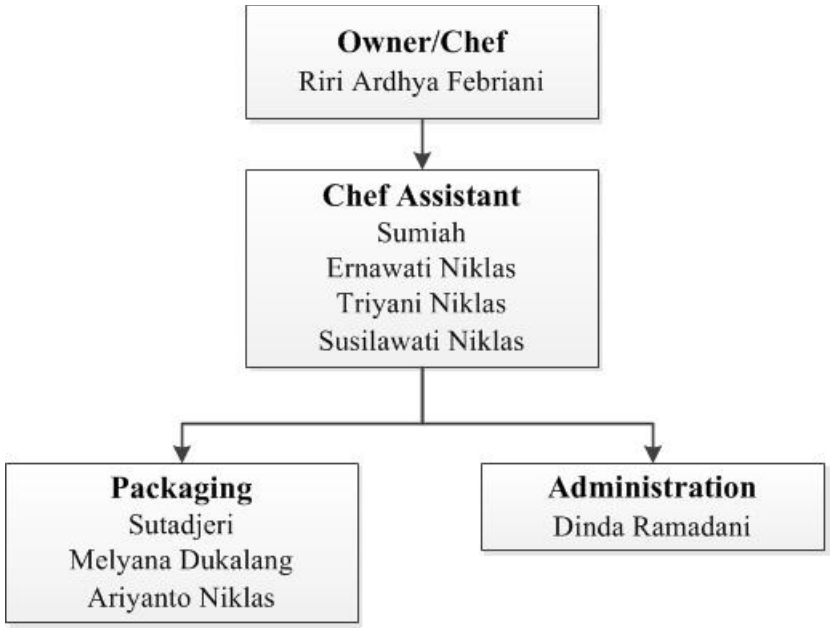
1. Menjaga kualitas dan kebersihan produk yang dihasilkan.
2. Memberikan pelayanan yang baik bagi konsumen.
3. Mendengarkan saran dan keluhan konsumen dan mencoba memperbaiki kesalahan yang ada.
4. Menciptakan kreasi produk-produk baru.
5. Menggunakan bahan-bahan yang halal, berkualitas, dan aman untuk dikonsumsi, namun harga jual tetap terjangkau dan rasa tetap enak.
6. Menata, membungkus produk kiriman sebaik mungkin.

3.1.3. Struktur Organisasi Perusahaan

Struktur organisasi mempunyai arti penting karena struktur organisasi merupakan bentuk atau pola formal kegiatan dan hubungan antara berbagai bagian-bagian didalam suatu peusahaan. Dengan mengetahui struktur organisasi

dapat diperoleh gambaran tentang bagian-bagian yang ada di dalamnya, apa peranan masing-masing bagian tersebut dan wewenang serta tanggung jawabnya dalam melaksanakan tugasnya.

Berikut ini merupakan struktur organisasi pada toko kue Chérie Cake's.



Gambar 3.1. Struktur Organisasi Chérie Cake's

3.1.4. Deskripsi Tugas

Berdasarkan struktur organisasi yang tertera, berikut ini akan diuraikan tugas, wewenang dan tanggung jawab dari masing-masing unit dalam struktur organisasi tersebut sebagai berikut:

1. *Owner/Chef*

   Bertugas dalam mengontrol seluruh kegiatan dari proses pembuatan kue.dan mengelola data pemesanan di server secara berkala, dan juga

mengolah seluruh proses pembuatan kue dari mulai menakar bahan, mengocok adonan, hingga memberikan topping.

2. *Chef Assistant*

   Memiliki peran dalam membantu *Chef* yaitu, antara lain mengolah seluruh proses pembuatan kue dari mulai menakar bahan, mengocok adonan, hingga memberikan topping.

3. *Packaging*

   Bagian *Packaging* bertugas dalam menata kue yang telah siap untuk dikemas di dalam toples atau dus.

4. *Administration*

   Mengelola pencatatan keuangan dari pemasukan dan pengeluaran untuk bahan baku dan mengelola penjualan.

3.2. Metode Penelitian

Pada subbab ini penulis akan menerangkan metode penelitian yang akan digunakan dalam penelitian ini. Metode penelitian ini meliputi desain penelitian, jenis dan metode pengumpulan data, serta metode pendekatan dan pengembangan sistem. Desain penelitian yang digunakan oleh penulis adalah Deskriptif dan *Action Research*. Sedangkan jenis pengumpulan data menggunakan data primer, dan data sekunder. Metode pendekatan sistem yang penulis pilih untuk perancangan aplikasi ini adalah pendekatan berorientasi objek dan metode pengembangan sistem yang digunakan adalah *Rational Unified Process (RUP)*.

3.2.1. Desain Penelitian

Penelitian ini menggunakan metode Deskriptif dan *Action Research*. Penelitian deskriptif dapat diartikan sebagai proses pemecahan masalah yang diselidiki dengan melukiskan keadaan subyek dan obyek pada saat sekarang berdasarkan fakta-fakta yang ada. Dimana dalam metode ini menggambarkan semua data yang kemudian dianalisis dan dibandingkan berdasarkan kenyataan yang sedang berlangsung dan selanjutnya mencoba untuk memberikan pemecahan masalahnya.

Pelaksanaan metode penelitian deskriptif tidak terbatas sampai pada pengumpulan dan penyusunan data, tetapi meliputi analisis dan interpretasi tentang data tersebut, selain itu semua yang dikumpulkan memungkinkan menjadi kunci terhadap apa yang diteliti.

Penelitian ini menggunakan metode *Action Research* (penelitian tindakan). Metode *Action Research* atau tindakan merupakan penelitian langsung, disertai dengan praktek di lapangan. Membuat suatu program yang akan dilaksanakan secara sistematis dan terencana, serta mempunyai nilai perbaikan yang signifikan. Penelitian tindakan ini lebih efektif, karena akan terlihat langsung hasilnya. Salah satu syarat dalam melakukan penelitian tindakan adalah adanya keinginan dari orang yang memilki masalah untuk mengidentifikasi masalah yang ada dan mempunyai keinginan untuk memecahkannya.

3.2.2. Jenis dan Metode Pengumpulan Data

Metode pengumpulan data merupakan suatu teknik atau cara untuk memperoleh, mencari, mengumpulkan dan mencatat data yang di gunakan untuk menyusun karya . Adapun pengumpulan data itu sendiri terbagi menjadi dua yaitu data primer dan data sekunder.

3.2.2.1. Sumber Data Primer

Data primer adalah data yang hanya dapat kita peroleh dari sumber asli (nara sumber) / responden penelitian baik melalui pengamatan maupun pencatatan terhadap objek penelitian.

Dalam penelitian ini penulis mengumpulkan data primer dengan menggunakan teknik :

1. Pengamatan Langsung (Observasi)

Observasi merupakan suatu cara pengumpulan data melalui pengamatan dan pencatatan oleh pengumpul data terhadap gejala atau peristiwa yang diselidiki pada obyek penelitian secara langsung tetapi tidak ada interaksi dengan objek yang diteliti dengan cara melakukan tinjauan langsung ke lapangan.

Penulis melakukan tinjauan langsung ke lapangan untuk mencari data-data yang diperlukan terhadap objek yang diteliti dan melihat secara langsung kegiatan yang dimulai pemesanan hingga pengambilan pemesanan.

2. Wawancara

Wawancara merupakan proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambil bertatap muka antara si penanya atau pewawancara dengan si penjawab atau responden. Dalam penelitian ini penulis melakukan wawancara secara langsung dengan mengajukan berbagai pertanyaan kepada pihak yang ikut terlibat langsung.

3.2.2.2. Sumber Data Sekunder

Data sekunder adalah data yang digunakan untuk mendukung data primer, merupakan jenis data yang sudah diolah terlebih dahulu oleh pihak pertama, data sekunder diambil secara tidak langsung dari objek penelitian misalnya data ini diperoleh dari buku-buku, jurnal, tutorial, internet dan lain-lain.

Dokumentasi adalah teknik pengumpulan data dengan cara mengumpulkan dokumen-dokumen yang berhubungan dengan objek penelitian. Dalam hal ini, dokumen yang diperoleh akan dianalisis agar diperoleh data yang sesuai dengan penelitian.

3.2.3. Metode Pendekatan dan Pengembangan Sistem

Dalam merancang suatu sistem dibutuhkan sebuah metode pendekatan dan pengembangan sistem untuk membantu dalam proses perancangan dan pembuatannya. Berikut adalah penjelasan metode pendekatan dan pengembangan sistem yang akan digunakan.

3.2.3.1. Metode Pendekatan Sistem

Pada dasarnya saat ini pengembangan sistem dapat dikategorikan dalam 2 pendekatan pengembangan, yaitu pengembangan secara terstruktur daan pengembangan secara *object oriented* (berorientasi objek). Metode pendekatan yang penulis ambil adalah pendekatan berorientasi objek yang divisualisasikan dengan UML.

Dalam jurnal yang ditulis oleh Hendrik, Fathul Wahid yang berjudul "Pengembangan Learning Management System" menerangkan bahwa :

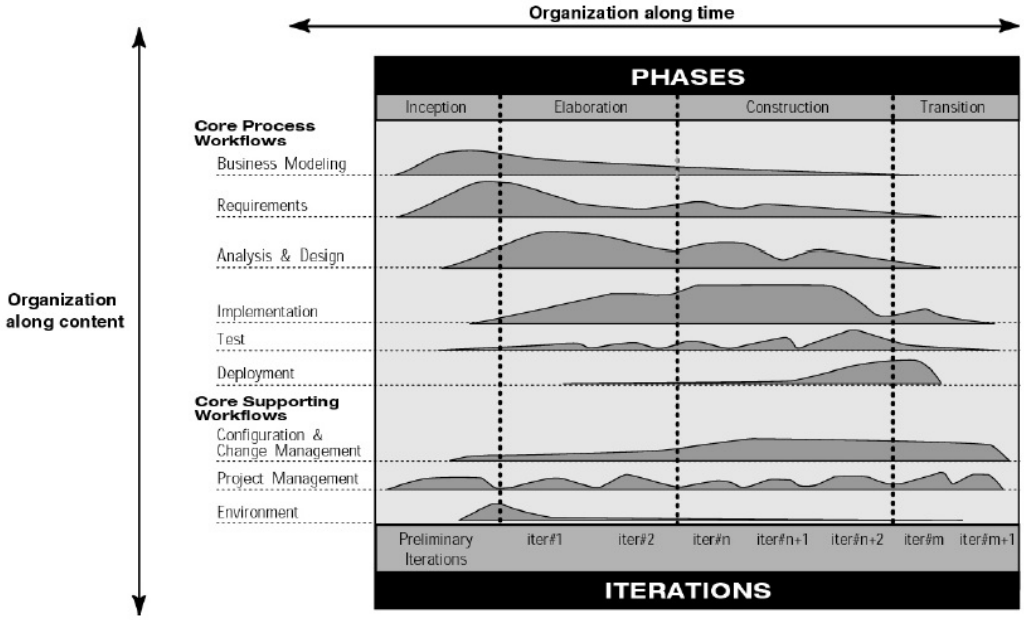
"Metode berorientasi obyek (*object oriented method*) merupakan suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan abstraksi obyek-obyek yang ada di dunia nyata. Brooks (1987) – dalam Nugroho (2002) – menyatakan bahwa bagian tersulit dari pengembangan perangkat lunak dan / atau sistem informasi adalah tahap analisis, yang didalamnya abstraksi dilakukan. Dalam metode berorientasi obyek kejadian atau hubungan antar entitas dalam dunia nyata direpresentasikan dalam obyek-obyek. "

3.2.3.2. Metode Pengembangan Sistem

Metode pengembangan yang penulis pakai adalah metode *Rational Unified Process* (RUP). *Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

1. **Dimensi pertama** digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction,* dan *Transition.*
2. **Dimensi kedua** digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how* dan *when*. Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Manegement, Project Management, Environtment.*



Gambar 3.2. Arsitektur *Rational Unified Process* (RUP)

(Sumber : *Rational-The Software Development Company, Rational Unified Process: Best Practices for Software Development Teams*)

Fase-fase dalam *Rational Unified Process* :

Berdasarkan pada Gambar 3.2. di atas, menjelaskan mengenai fase-fase pada *Rational Unified Process* yang termasuk ke dalam dimensi pertama dari RUP :

a. ***Inception*** (Insepsi)

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan *user*, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*). Pada akhir tahap ini, prototipe perangkat lunak versi *Alpha* harus sudah dirilis. Kegiatan yang dilakukan pada tahap ini diantaranya :

1) Menentukan ruang lingkup proyek.

Ruang lingkup proyek meliputi, tata cara untuk menentukan waktu proyek dimulai, perencanaan lingkup proyek yang akan dikerjakan, pendefinisian ruang lingkup proyek, verifikasi proyek serta kontrol atas perubahan yang mungkin terjadi saat proyek tersebut dimulai.

2) Membuat *Business Case*.

Business Case adalah alat yang mendukung perencanaan dan pengambilan keputusan-termasuk keputusan tentang apakah yang akan dibeli, produk apa yang akan dibawa ke pasar, proyek apa yang akan di danai, dan vendor mana yang akan dipilih, atau kapan proyek tersebut dilaksanakan. *Business Case* umumnya dirancang untuk menjawab pertanyaan seperti di atas tentang konsekuensi dari suatu tindakan atau keputusan.

3) Menjawab pertanyaan "apakah yang dikerjakan dapat menciptakan 'good business sense' sehingga proyek dapat dilanjutkan.

b. ***Elaboration*** (Elaborasi)

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilisan prototipe versi *Betha* dari perangkat lunak. Kegiatan yang dilakukan pada tahap ini diantaranya :

1) Menganalisa berbagai persyaratan dan resiko

Di dalam sebuah proyek, terdapat beberapa persyaratan-persyaratan yang harus dipenuhi agar suatu proyek tersebut dapat terlaksana dengan baik. Selain itu melakukan analisa terhadap resiko juga diperlukan agar pada saat resiko tersebut terjadi, sudah ada persiapan untuk menanggulangi resiko-resiko yang akan terjadi.

2) Menetapkan *baseline*

Baseline adalah sebuah *copy* dari suatu *project*. Kita dapat membandingkan antara *baseline* dengan kondisi yang sekarang untuk mengevaluasi *progress*.

3) Merencanakan fase berikutnya yaitu *Construction*

Merupakan titik awal perancangan dan pengembangan *testing* dan dapat dipakai untuk menganalisa iterasi.

c. ***Construction*** (Konstruksi)

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Tujuan utama adalah untuk membangun sistem perangkat lunak. Pada tahap ini, fokus utama adalah pada pengembangan komponen dan fitur lain dari sistem yang dirancang. Ini adalah tahap ketika sebagian besar terjadi pengkodean. Kegiatan yang dilakukan pada tahap ini diantaranya :

1) Melakukan sederetan iterasi

Melakukan serangkaian kegiatan dan tugas, dengan sumber daya yang ditetapkan, mengandung ketergantungan tugas, untuk iterasi. Biasanya terdapat 2 rencana iterasi yang aktif pada setiap waktu yang sama, yaitu :

a) Rencana iterasi yang saat ini digunakan untuk melacak kemajuan dalam iterasi yang sedang berlangsung.
b) Rencana iterasi yang selanjutnya adalah rencana yang digunakan untuk merencanakan iterasi selanjutnya. rencana ini di siapkan untuk menghadapi akhir dari iterasi saat ini.

2) Pada setiap iterasi akan melibatkan proses berikut : analisa desain, implementasi dan *testing*.

d. ***Transition*** (Transisi)

Instalasi, *deployment*, dan sosialisasi perangkat lunak dilakukan pada tahap ini. Tujuan utama adalah untuk transisi sistem dari ke pengembangan produksi, membuatnya tersedia untuk dan dipahami oleh pengguna akhir. Kegiatan ini meliputi pelatihan tahap akhir pengguna dan pengelola dan beta *testing* dari sistem untuk memvalidasi sistem tersebut terhadap pengguna akhir. Kegiatan yang dilakukan pada tahap ini diantaranya :

1) Membuat sistem yang sudah dimodelkan menjadi suatu produk jadi.
2) Dalam fase ini dilakukan :
    a) *Beta* dan *performance testing*

        Melakukan *testing* pada produk yang sudah jadi untuk mengetahui apakah masih terdapat *bug* atau tidak.
    b) Membuat dokumentasi tambahan seperti : *training*, *user guides* dan *sales kit*.
    c) Membuat rencana peluncuran produk ke komunitas pengguna.

Berikut adalah penjelasan mengenai langkah-langkah *workflow* pada *Rational Unified Process* yang termasuk ke dalam dimensi kedua dari RUP :

a. ***The Business Modeling Workflow***

Model bisnis menjelaskan visi organisasi dimana sistem akan diturunkan dan bagaimana kemudian visi ini digunakan sebagai dasar untuk menguraikan proses, peran dan tanggung jawab. Tujuan dari model bisnis adalah untuk membangun pemahaman yang lebih baik serta sebagai saluran komunikasi antara teknik bisnis dan *software engineering*. Objek dari *workflow* ini sama dengan metodologi lainnya,

tetapi pada RUP teknik yang sama digunakan sebagai langkah selanjutnya dalam pengembangan, untuk meyakinkan proses *end to end* dan bahwa setiap orang berbicara dalam bahasa yang sama. *Business Modeling* menggambarkan kepada kita proses bisnis dengan menggunakan *business use case* yang bertujuan untuk meyakinkan suatu pemahaman umum antar semua *stakeholders* dari apa yang proses bisnis perlukan untuk mendukung organisasi tersebut. *Business use case* dianalisa untuk memahami bagaimana bisnis seharusnya mendukung proses bisnis itu.

Fase-fase yang terlibat dalam *business modeling* :

1) Inception

Pertama kalinya *business modeling* dideklarasikan dan didefinisikan.

2) Elaboration

Peninjauan kembali terhadap *requirement* bisnis untuk meminimalisasikan terjadinya perubahan pada tahap selanjutnya yaitu *construction*.

3) Construction

Penerapan dari *business modeling* yang telah terdefinisi dalam bentuk *coding*.

4) Transition

Dimungkinkan apablia terjadi kesepakatan antara *developer* dengan *end users* dalam perawatan *software* yang telah dibuat.

b. The Requirements Workflow

Objek pada tahap ini menyusun sistem apa yang seharusnya ada dan mengapa perlu dibuat, mendefinisikan batas dari sistem, melihat kemungkinan ancaman keamanan serta bagaimana cara penanggulangannya, dan mengestimasi biaya dan skala waktu yang rumit. Visi dari sistem dibangun yang kemudian diterjemahkan kedalam *use*

case model dengan tambahan spesifikasi kebutuhan. Baik kebutuhan fungsional dan nonfungsional dikumpulkan dan di analisis.

Fase-fase yang terlibat antara lain :

1) Inception

Requirement dari *software* pertama kali dibahas. Lebih terfokus pada *requirement* pengembangan *software* yang akan dipakai.

2) Elaboration

Mengurangi / meninjau kembali *requirement* dari *software*, dan dimungkinkan terjadi pergantian *requirement* dalam *software* yang akan dikembangkan.

3) Construction

Perwujudan *requirement* yang ada dalam bentuk *coding* dari *software* yang dikembangkan beserta pengujian apakah *software* sudah memenuhi *requirement* awal.

4) Transition

Bisa saja *requirement* dalam fase ini berupa *requirement* dari *end users* untuk menambah aplikasi *software*, atau mungkin perawatan *software*, atau mungkin yang lain juga.

c. The Analysis and Design Workflow

Pada tahap ini requirements dari tahap dua diubah kedalam *implementation specification*. Analisis meyakinkan bahwa *functional requirements* ditemukan, secara khusus mengabaikan *requirements nonfungsional* dan *runtime environment*. Desainnya mengambil *output* dari analisis dan mengadaptasikannya kedalam pembatasan arsitektur dan *requirements* nonfungsional. Meliputi aktifitas pendefinisian dan penyaringan arsitektur, menganalisa perilaku, desain komponen

dan desain *database*. Tujuan dari analisis dan desain adalah menunjukkan bagaimana sistem akan terwujud.

Fase-fase yang terlibat :

1) Inception

Analisis dan desain sudah mulai dibahas dengan adanya pembahasan tentang *business modeling* dan *requirement.*

2) Elaboration

Fase inilah yang menjadi pusat perkembangan dari analisis dan desain. Perancangan dan analisa dilakukan pada fase ini.

3) Construction

Pada tahap ini *project* dikembangkan dalam bentuk *coding*.

4) Transition

d. The Implementation Workflow

Workflow mengkonversikan desain ke dalam implementasi. Kegiatannya meliputi merencanakan proses, mengkonversikan kelas dan objek dari tahap tiga ke dalam komponen, menguji komponen individual, dan membangun versi operasional dari sistem, dikenal sebagai '*the builds*'. Tujuan dari pelaksanaan ini adalah untuk menguji komponen-komponen yang dikembangkan sebagai unit dan untuk mengintegrasikan hasil yang diproduksi oleh individu pelaksana (atau tim) ke dalam sistem yang dapat dieksekusi.

Fase-fase yang terlibat :

1) Inception

Di tahap ini implementasi berlaku dengan terjadinya percakapan antara *end users* dan *developer* mengenai *software* yang akan dikembangkan.

2) Elaboration

Selain implementasi terhadap pembuatan *use case*, tahap ini juga memuat implementasi dari perkembangan perencanaan arsitektural dan sebagainya.

3) Construction

Pada fase ini implementasi terhadap rancangan *software* dan sebagainya diterapkan.

4) Transition

Implementasi yang terjadi pada tahap ini adalah penyerahan *software* terhadap *end users* dan implementasi pada penerapan aplikasi *software* yang telah dikembangkan.

e. The Test Workflow

Tahap ini menguji dan memverifikasi interaksi komponen, semua requirements telah diimplementasikan, dan kualitas produk yang telah dikembangkan dari ketiadaan kerusakan dan kemampuan untuk mencapai tujuan. Tujuan dari *test workflow* adalah untuk memverifikasi interaksi antara obyek, memverifikasi integrasi yang tepat dari semua komponen perangkat lunak, memastikan bahwa semua persyaratan telah benar dilaksanakan, mengidentifikasi dan memastikan bahwa cacat yang ditujukan sebelum penggelaran perangkat lunak.

Fase-fase yang terlibat :

1) Inception

Dalam fase ini testing dilakukan apabila *business modeling* dan *requirement* telah teridentifikasi. Testing dilakukan dengan tujuan menghasilkan kesepakatan antara *end users* dengan *developer*.

2) Elaboration

Testing di sini merupakan *testing* setelah *use case* diimplementasikan, masih seputar tercapainya kesepakatan antara *end users* dengan *developer*.

3) Construction

Testing sering kali dilakukan di akhir fase *construction*, karena *testing* dilaksanakan setelah penyelesaian program.

4) Transition

Testing dilakukan sebelum penyerahan *software* kepada *end users* dengan keadaan yang sebenarnya.

f. The Deployment Workflow

Tahap ini menyebarkan software yang telah selesai kepada user dan meliput:

a. Menguji *software* dalam *setting* operasional

b. Training the end users

c. Migrasi dari *software* yang sudah ada

d. Pengemasan *software*

e. Instalasi *software*

Tujuan dari penyebaran adalah untuk berhasil menghasilkan produk rilis, dan untuk memberikan perangkat lunak kepada pengguna akhir. Ini mencakup berbagai kegiatan termasuk rilis eksternal memproduksi perangkat lunak, kemasan perangkat lunak dan aplikasi bisnis, mendistribusikan perangkat lunak, menginstal perangkat lunak, dan memberikan bantuan dan bantuan kepada pengguna. Meskipun kegiatan penyebaran kebanyakan berpusat pada fase transisi, banyak kegiatan yang harus disertakan dalam fase-fase awal untuk mempersiapkan pengiriman pada akhir fase konstruksi.

Fase-fase yang terlibat :

1) Elaboration

Pengembangan tentang realitas dari *software* itu akan seperti apa dimulai pada fase ini.

2) Construction

Dalam fase ini pengembangan *software* secara nyata terjadi dengan adanya *coding*.

3) Transition

Fase yang paling berpengaruh karena adanya penyerahan *software* dari *developer* kepada *end users*.

g. The Configuration and Change Management Workflow

Tahap ini menjalankan dan merawat integritas dari proyek. Kegiatannya meliputi memonitor dan mengatur perubahan permintaan, perubahan biaya, dan tetap mengontrol berbagai versi produk. Juga meliputi manajemen konfigurasi *hardware* dan *software*.

Fase-fase yang terlibat :

1) Inception

Terjadi diskusi mengenai konfigurasi dari sistem software yang diinginkan.

2) Elaboration

Masih membahas seputar konfigurasi *software*, ditambah dengan perubahan-perubahan yang terjadi, terkait dengan diminimalisasikannya perubahan dalam fase selanjutnya.

3) Construction

Dalam fase inilah akan terlihat jelas penerapan dari konfigurasi yang telah ditentukan, dan mungkin tidaknya konfigurasi yang diinginkan terpenuhi.

4) Transition

Konfigurasi yang ada adalah mengenai konfigurasi sistem dalam keadaan yang sesungguhnya.

h. The Project Management Workflow

Tahap ini menyediakan *framework* untuk memanajemen *software* dan memanajemen resiko. Tahap ini juga menyediakan pedoman untuk *planning*, *staffing*, *monitoring* dan secara umum menunjukan manajemen proyek. Semua fase di sini di gunakan.

i. The Environment Workflow

Tahap ini menjelaskan tentang pendukung proyek dengan proses yang relevan, metode-metode, dan *tools* dalam organisasi. Semua fase di sini di gunakan.

3.2.3.3. Alat Bantu Analisis dan Perancangan

Pada perancangan sistem informasi akademik ini, penulis menggunakan metode pendekatan berorientasi objek, terdapat banyak alat bantu yang dapat digunakan dalam analisis dan desain berorientasi obyek menyebabkan munculnya ketidakjelasan alat bantu apa yang paling unggul. Para pengguna harus memilih diantara bahasa pemodelan dan alat bantu pemodelan yang beragam. Untuk mengatasi masalah tersebut maka *Object Management Group* (OMG) kemudian mengeluarkan UML, dimana dengan adanya UML ini diharapkan dapat

mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkugan industri perangkat lunak.

1. *Use Case Diagram*

Diagram *use case* menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar.

Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case* diagram menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case* diagram.

Dalam diagram *use-case* terdapat beberapa hubungan antar *use-case* antara lain: *Extend* dan *Include*. Hubungan *include* menunjukkan bahwa sebuah *business use-case* selalu termasuk dalam alur kerja dari *business use-case* lainnya. Hubungan *extend* menunjukkan bahwa sebuah *use-case* merupakan kelanjutan dari *business use-case* lainnya jika suatu syarat tertentu telah dipenuhi.

2. *Activity Diagram*

Activity diagram (diagram aktivitas) adalah diagram *flowchart* yang disempurnakan. Diagram aktivitas menggambarkan operasi pada suatu obyek atau proses pada sebuah organisasi. Kelebihan diagram aktivitas dibandingkan dengan diagram *flowchart* adalah adanya dukungan konkurensi (pelaksanaan aktivitas

secara bersamaan), pengiriman pesan dan *swimlane* (pelaku/penanggung jawab aktivitas).

Diagram aktivitas sangat berguna ketika kita ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Diagram aktivitas diawali dengan lingkaran hitam, dan diakhiri dengan lingkaran hitam bertepi putih. Aktivitas digambarkan dengan kotak persegi panjang bersudut lengkung. Setiap aktivitas dihubungkan dengan panah dari awal hingga akhir diagram aktivitas.

Sama halnya dengan diagram *flowchart*, diagram aktivitas pun memiliki simbol yang sama untuk menggambarkan keputusan. Keputusan digambarkan dengan intan, namun deskripsi kondisi yang menyertai keputusan diletakkan diluar simbol intan. Diagram aktivitas dapat menggambarkan konkurensi, yaitu satu atau lebih aktivitas yang berjalan secara bersamaan. Konkurensi diawali dengan sebuah garis tebal horizontal yang menjadi tempat keluarnya garis aktivitas. Konkurensi juga diakhiri dengan garis tebal horizontal.

3. *Sequence Diagram*

Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sebuah *Sequence diagram*, secara khusus, menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sebuah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *use case*.

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu.

4. *Class Diagram*

Diagram *class* menggambarkan seluruh *class* yang terlibat dalam sebuah perangkat lunak. Diagram ini menggambarkan seluruh atribut dan operasi suatu *class* beserta hubungannya dengan *class* lainnya.

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class Diagram* juga menunjukkan *property* dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

5. *Object Diagram*

Object diagrams memodelkan pandangan statik terhadap sistem untuk memodelkan struktur objek.

Menurut Bambang Heriyanto (2004:291) *Object Diagram* adalah diagram instan (*instance diagram*) yang mendeskripsikan instan-instan kelas.

Pemodelan struktur objek melibatkan *snapshot* dari objek-objek sistem pada suatu waktu. *Object diagrams* merepresentasikan satu *frame* statik papan cerita dinamis dari diagram interaksi.

Kegunaan *object diagram* adalah mendeskripsikan bagaimana kumpulan objek tertentu saling berhubungan. *Object diagrams* adalah diagram instan yang

mendeskripsikan instan-instan kelas. Diagram instan berguna untuk dokumentasi skenario, serta kasus pengujian dan mendiskusikan contoh-contoh instanisasi *class diagrams.*

6. *Component Diagram*

Diagram komponen menggambarkan alokasi semua kelas dan obyek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software,* seperti *soure code, binary code* dan komponen tereksekusi.

7. *Deployment Diagram*

Diagram *deployment* menggambarkan bagaimana konfigurasi perangkat keras dimana sistem akan diinstalasikan. Diagram ini memberi simbol untuk segala bentuk perangkat keras, seperti: komputer, printer, router, hub, jaringan, dan sebagainya.

Diagram *deployment* menunjukkan tata-letak perangkat keras secara fisik, dan komponen yang terdapat didalamnya. Diagram ini menunjukkan secara lengkap arsitektur dari sistem dan perangkat kerasnya. Diagram ini akan memperjelas pengguna bagaimana hasil akhir dari sistem yang dikembangkan, dan mempermudah teknisi dalam pemeliharaan sistem.

Deployment diagram bisa juga berisi komponen-komponen, masing-masing komponen itu berada di suatu node. Juga dapat berisi paket atau subsistem yang

digunakan mengelompokkan elemen-elemen di model menjadi potongan-potongan.

3.3. Pengujian Software

Penulis akan menggunakan teknik *black box testing* dalam pengujian *software*. Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancang data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dijalankan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Pada pengujian *black box*, mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Teknik pengujian *black box* juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *use case* dan informasi analisis yang lain.

Beberapa jenis kesalahan yang dapat di identifikasi :

1. Fungsi tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan pada struktur data (pengaksesan basis data)
4. Kesalahan inisialisasi dan akhir program
5. Kesalahan performasi.