

**ANALISIS PENGGUNAAN MODEL SOFTWARE DEVELOPMENT LIFE
CYCLE WATERFALL DALAM PENGEMBANGAN PERANGKAT LUNAK**

(Studi Kasus PT. Rabiha Pilar Informatika Yogyakarta)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika



Disusun Oleh :

Nama : Radix Megantoro

No. Mahasiswa : 09523247

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2013

LEMBAR PENGESAHAN PEMBIMBING
ANALISIS PENGGUNAAN MODEL SOFTWARE DEVELOPMENT LIFE
CYCLE WATERFALL DALAM PENGEMBANGAN PERANGKAT LUNAK
(STUDI KASUS PT. RABIHA PILAR INFORMATIKA YOGYAKARTA)

TUGAS AKHIR

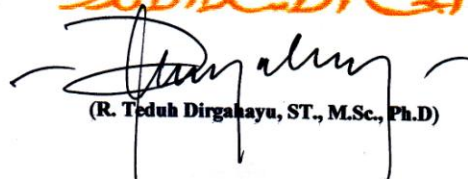
Disusun Oleh:

Nama : Radix Megantoro
No. Mahasiswa : 09523247



Yogyakarta, Juli 2013

Pembimbing


(R. Teduh Dirgahayu, ST., M.Sc., Ph.D)

LEMBAR PENGESAHAN PENGUJI
ANALISIS PENGGUNAAN MODEL SOFTWARE DEVELOPMENT LIFE
CYCLE WATERFALL DALAM PENGEMBANGAN PERANGKAT LUNAK
(STUDI KASUS PT. RABIHA PILAR INFORMATIKA YOGYAKARTA)

TUGAS AKHIR

Disusun oleh:

Nama : Radix Megantoro

No. Mahasiswa : 09523247

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk
Memperoleh Gelar Sarjana Teknik Informatika

Fakultas Teknologi Industri, Universitas Islam Indonesia

Yogyakarta, Juli 2013

Tim Penguji,
Ketua

R. Teduh Dirgahayu, ST., M.Sc., Ph.D.

Anggota 1

Yudi Prayudi, S.Si, M.Kom

Anggota II

Ahmad Luthfi, M.Kom



Mengetahui,
Ketua Program Studi Teknik Informatika
Universitas Islam Indonesia



(Yudi Prayudi, S.Si, M.Kom)

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertanda tangan dibawah ini,

Nama : Radix Megantoro

No. Mahasiswa : 09523247

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 5 Juli 2013

Radix Megantoro

HALAMAN PERSEMBAHAN

Alhamdulillah Rabbil 'Aalamiin

Puji dan syukur saya panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya serta memberikan kemudahan sehingga saya dapat menyelesaikan tugas akhir ini dengan baik, semoga bermanfaat dikemudian hari.

Dengan izin-Nya pula, saya persembahkan karya saya ini untuk :

Bapak dan Mama. Yang dengan keikhlasan dan kesabaran tiada henti - hentinya memberikan dukungan dan semangat, serta doa untuk keberhasilan saya, mudah-mudahan ini merupakan suatu bentuk dari sebagian kecil bakti dan patuh saya. Jasa-jasa kalian takkan pernah tergantikan.

Keluargaku, Taufan Meganda, Tante Kris, Tante Yani, Om In, Tia, Icha Kakak Lita. Yang selalu memberi dukungan dan motivasi semangat.

Terima kasih kepada Rama dan Iqbal yang menemani perjalanan wawancara, terima kasih pula kepada narasumber mas Wisnu, mas Fahry, mas Eddy, mas Hari atas kesediaan informasi yang diberikan

Keluarga Kontrakan, Ully, Iqbal, Alan, Zohan, Firman, Faisal, Kevin, Rama, Wempi, dan Pak Is sekeluarga. Begitu banyak kenangan indah bersama kalian.

Keluarga Kos Amanah, Aan, Salis, Eko, Rizeki, Fauzan. Terima kasih atas semua support yang kalian berikan.

Pak Teduh, terima kasih atas ide dan masukan bagi tugas akhir saya dan juga yang telah membimbing dalam pengerjaan tugas akhir.

My beloved Infinity, yang gak mungkin kusebutkan satu-satu, bersama kalian aku temukan keluarga di Yogyakarta ini.

Untuk teman-temanku KKN Unit Unit 21 : mbak Ari, Feby, Inung, Budy, mas Bro, Fajar, Saifu. Terimakasih atas kebersamaannya selama ini dan momen KKN yang takkan terlupakan olehku seumur hidup.

Kepada teman-teman FTI UII angkatan 2007, 2008, 2009, 2010, 2011 dan 2012 yang tak mungkin saya sebutkan satu persatu disini, terimakasih atas kebersamaannya selama ini.

Dan semuanya yang telah mendukung, maaf apabila ada yang tidak disebutkan.

HALAMAN MOTTO

Sesungguhnya manusia itu benar-benar berada dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran.

(QS Al Ashr ayat 2, 3)

Adapun orang yang memberikan dan bertakwa, dan membenarkan adanya pahala yang terbaik, maka kami kelak akan menyiapkan baginya jalan yang mudah.

(QS Al Lail ayat 5, 6, 7)

Orang yang tidak sekolah saja bisa mendapat nasib yang baik, apalagi kita yang sekolah, yakin masa depan akan cerah

(Fahry Bafadhal)

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya, maka penulis dapat menyelesaikan Tugas Akhir ini.

Rangkaian penyelesaian yang telah penulis lakukan tidak lepas dari bimbingan dan dorongan banyak pihak. Maka penghargaan yang tinggi dan ucapan terima kasih yang sebesar-besarnya penulis tujukan kepada :

1. Bapak Gumbolo Hadi Susanto, Ir., M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi, S.Si., M.Kom, selaku ketua jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia
3. R. Teduh Dirgahayu, ST., M.Sc., Ph.D. selaku dosen pembimbing yang telah bersedia meluangkan waktu, tenaga dan pikirannya membimbing penulis dalam menyelesaikan Tugas Akhir.
4. Bapak (Sinom Pradopo), Mama (Rufaidah), Taufan Megananda keluarga tercinta yang tak henti hentinya memberikan dorongan semangat dan doanya.
5. Teman-teman kos dan kontrakan, Wempi, Alan, Iqbal, Aan, Salis, Zeki, Faisal, Zohan, Fauzan, Kevin, Rama, Affan yang telah banyak membagikan ilmu, pengalaman, rasa kekeluargaan dan tanggung jawab kepada penulis.
6. Teman-teman jurusan Teknik Informatika UII (INFINITY '09) dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu.
7. Seluruh teman-teman di Teknik Informatika UII satu kata yang selalu saya camkan "SATU UNTUK SEMUA SEMUA UNTUK SATU". Terimakasih untuk semuanya.

Penulis menyadari sepenuhnya bahwa Laporan Tugas Akhir ini masih memiliki kelemahan dan kekurangan. Untuk itu penulis mengharapkan saran dan kritik yang membangun agar menjadi lebih baik.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 05 Juli 2013

Radix Megantoro

SARI

Perangkat lunak adalah salah satu hasil dari teknologi yang mengalami perkembangan paling pesat. Saat ini telah banyak perangkat lunak yang diciptakan dan dipakai dalam kehidupan sehari-hari. Dalam pengembangan perangkat lunak terdapat tahapan-tahapan yang harus dilalui. Tahapan-tahapan tersebut dijadikan pedoman oleh para pengembang perangkat lunak yang disebut SDLC (*Systems/Software Development Life Cycle*). Tahapan umum dalam SDLC adalah, analisis, desain, *coding*, implementasi dan perawatan.

SDLC memiliki jenis yang berbeda-beda. SDLC yang paling tua adalah model *waterfall* yang mana tahapan proses pengembangan perangkat lunaknya digambarkan seperti air terjun. SDLC ini disebut pula “*linear sequential model*”. Model *waterfall* tidak mengizinkan adanya proses yang kembali ke proses sebelumnya jika proses selanjutnya sudah berjalan. Penelitian ini bertujuan untuk menunjukkan penggunaan model *waterfall* dalam usaha pengembangan perangkat lunak khususnya pada perusahaan PT. Rabiha Pilar Informatika di Kota Yogyakarta.

Kata kunci : *Pengembangan perangkat lunak, SDLC, Model Waterfall.*

TAKARIR

<i>Software</i>	Perangkat Lunak
<i>SDLC (Systems/Software Development Life Cycle)</i>	Proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya
<i>Feasible</i>	Dapat dikerjakan
<i>Requirements</i>	Kebutuhan
<i>Design</i>	Desain
<i>High-level design</i>	Desain yang berisi pengembangan struktur asitektural untuk komponen perangkat lunak, <i>database</i> , <i>user interface</i> , dan lingkungan pengoperasian
<i>Low-level design</i>	Desain yang berisi perincian algoritma dan struktur data yang dibutuhkan untuk pengembangan perangkat lunak
<i>Coding</i>	Koding
<i>Testing</i>	Pengetesan
<i>Roll out / deployment</i>	Penyebaran/pemasangan
<i>Maintenance</i>	Perawatan

Overlapping

Melakukan pekerjaan dalam waktu yang bersamaan

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO.....	vi
KATA PENGANTAR	vii
SARI	ix
TAKARIR	x
DAFTAR ISI	xii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvi
BAB I	1
1.1. LatarBelakang.....	1
1.2. RumusanMasalah.....	2
1.3. BatasanMasalah.....	2
1.4. Tujuan	2
1.5. Manfaat	3
1.7. SistematikaPenulisan	3
BAB II.....	5
2.1. Perangkat Lunak (Software).....	5
2.2. <i>Sistems Development</i> Life Cycle (SDLC)	8
2.2.1. Definisi.....	8
2.2.2. Tahapan dalam SDLC.....	8

2.3.	Model SDLC <i>Waterfall</i>	10
2.3.1.	Keuntungan Model <i>Waterfall</i>	16
2.3.2.	Kerugian Model <i>Waterfall</i>	17
BAB III.....		18
3.1.	Penelitian Kualitatif	18
3.1.1.	Definisi dan Alasan Penggunaan Metode Kualitatif	18
3.2.	Subjek Penelitian	20
3.2.1.	Teknik Pemilihan Subjek	20
3.2.2.	Jumlah Subjek Penelitian	20
3.3.	Metode Pengumpulan Data	21
3.4.	Alat Bantu Pengumpulan Data	23
3.5.	Prosedur Penelitian	23
3.5.1.	Keterbatasan Penelitian.....	23
3.5.2.	Tahap Persiapan.....	23
3.5.3.	Tahap Pelaksanaan.....	24
3.5.4.	Tahap Analisis Data.....	25
3.5.5.	Tahap Penyajian Data	26
BAB IV		27
4.1.	Karakteristik Informan.....	27
4.2.	Hasil Wawancara	27
4.2.1.	Model <i>Waterfall</i> yang Digunakan	27
4.2.2.	Tahapan Model <i>Waterfall</i> PT. Rabiha Pilar Informatika.....	29
4.2.2.1.	Project Planning (Perencanaan Proyek)	30
4.2.2.2.	Analisis Kebutuhan dan Perancangan Sistem.....	32
4.2.2.3.	<i>Development</i> (Pengembangan)	36
4.2.2.4.	Testing (Pengetesan)	37
4.2.2.5.	Release (Peluncuran).....	39
4.2.2.6.	Maintenace (Perawatan)	40

4.2.3. Masalah Dalam Penggunaan Model <i>Waterfall</i>	40
4.3. Tabel Rangkuman Penelitian.....	41
4.4. Pembahasan.....	44
BAB V.....	52
5.1. Kesimpulan.....	52
5.2. Saran	52
DAFTAR PUSTAKA	53
LAMPIRAN	55

DAFTAR GAMBAR

Gambar 1. Tahapan SDLC secara umum.....	9
Gambar 2. Model SDLC <i>Waterfall</i>	14
Gambar 3. Alur Proses Model <i>Waterfall</i> PT. RPI	28
Gambar 4. Halaman Testing.....	38
Gambar 5. Model Waterfall PT. RPI	46
Gambar 6. Model Waterfall.....	47

DAFTAR TABEL

Tabel 1. Tabel Pengambilan Data.....	25
Tabel 2. Tabel Tema dan Sub Tema Hasil Wawancara.....	44

BAB I

PENDAHULUAN

1.1. LatarBelakang

Perangkat lunak adalah salah satu hasil teknologi yang mengalami perkembangan paling pesat dalam beberapa tahun. Dalam hal pengembangan sistem/perangkat lunak segala sesuatunya harus memiliki kerangka kerja (Nugroho, 2010). Kerangka kerja pengembangan sistem/perangkat lunak ini disebut pula sebagai *systems development life cycle* (SDLC). SDLC adalah proses untuk mengembangkan atau mengubah suatu system perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (A.S. & M.Shalahuddin, 2011). Dalam SDLC terdapat beberapa proses seperti perencanaan, pendefinisian sistem, analisis kebutuhan, perancangan, pengujian dan integrasi, implementasi, serta pengoperasian dan pemeliharaan (Nayak & Qiu, 2006). Terdapat beberapa model SDLC yang biasa digunakan, contoh *waterfall*, *spiral*, *incremental*, dan RAD (*Rapid Application Development*) (Schwalbe, 2009).

Karena banyaknya model SDLC yang dikenal, maka dalam pengembangan perangkat lunak selalu terjadi perbedaan dalam penggunaan model SDLC. Dari semua model SDLC yang ada, model *waterfall* adalah salah satu model yang dipelajari dan digunakan hingga sampai saat ini, karena model ini merupakan model dasar dari semua model SDLC yang lain. Model *waterfall* ini merupakan model yang paling mudah dipahami (Pries & Quigley, 2011), sehingga ada beberapa pengembang perangkat lunak masih mempelajari bahkan menggunakan model ini.

Namun model *waterfall* memiliki resiko paling tinggi, karena pada model ini pengembangan dilakukan tahap demi tahap. Satu tahap yang akan dikerjakan harus

menunggu tahap sebelumnya selesai atau berjalan berurutan (Andreas, 2007). Tidak ada proses yang berjalan mundur. Pada model ini, proses komunikasi dengan klien sangat minim sekali seperti yang dikatakan oleh Hari Setiaji, *Team Leader* dalam perusahaan RPI. Hal ini menjadikan model *waterfall* memiliki resiko yang tinggi, karena jika pada tahap analisis kebutuhan terdapat kesalahan atau kurangnya informasi yang dibutuhkan, maka proyek pengembangan perangkat lunak tersebut terancam gagal. Hal inilah yang menjadi alasan mengapa penelitian ini dilakukan, untuk menemukan bagaimana penggunaan model *waterfall* khususnya pada perusahaan PT. Rabiha Pilar Informatika dalam usaha pengembangan perangkat lunak di kota Yogyakarta.

1.2. RumusanMasalah

Berdasarkan latar belakang di atas maka didapatkan beberapa rumusan masalah sebagai berikut:

- Bagaimana penggunaan model *waterfall* pada perusahaan PT. Rabiha Pilar Informatika?
- Apa kendala yang biasanya ditemui saat menggunakan model SDLC *waterfall* ?

1.3. BatasanMasalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- Wilayah usaha pengembangan perangkat lunak hanya di kota Yogyakarta

1.4. Tujuan

Tujuan tugas akhir ini adalah sebagai berikut:

- Memberikan gambaran tentang penggunaan model *waterfall* serta kendala yang terjadi di PT. Rabiha Pilar Informatika dalam menerapkan model *waterfall* dalam melakukan pengembangan perangkat lunak.

1.5. Manfaat

Manfaat tugas akhir ini adalah sebagai berikut:

- Dapat memberikan gambaran yang lebih nyata dalam penggunaan model SDLC *waterfall* sebagai acuan dalam menyelesaikan proyek pengembangan perangkat lunak.
- Institusi pendidikan dapat lebih memberikan gambaran tentang penerapan model *waterfall* di perusahaan pengembangan perangkat lunak secara nyata kepada anak didik mereka.

1.6. Metodologi Penelitian

Metode dalam penelitian ini adalah:

- Metode wawancara, merupakan pengumpulan data secara langsung dengan melakukan tanya jawab dengan PT. Rabiha Pilar Informatika.
- Metode studi pustaka, merupakan cara pengumpulan berbagai informasi dari sumber kepustakaan, seperti buku dan sumber online.

1.7. Sistematika Penulisan

Tugas akhir ini disusun dalam beberapa bagian dengan sistematika tertentu, untuk memudahkan pembaca memahami isinya. Berikut adalah penjelasan bagian-bagian dalam tugas akhir ini:

Bab 1. Pendahuluan

Berisi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta sistematika penulisan.

Bab 2. Tinjauan Pustaka

Berisi teori dasar mengenai penelitian ini. Pembahasan berkisar tentang definisi dan tipe penelitian yang dijalani.

Bab 3. Metodologi Penelitian

Berisi metode dan proses yang digunakan dalam melakukan penelitian ini.

Bab 4. Hasil dan Pembahasan

Berisi pembahasan hasil-hasil data dan informasi yang diperoleh yang sebagai mana telah dilaksanakan.

Bab 5. Kesimpulan dan Saran

Berisi kesimpulan hasil penelitian dan saran pengembangan penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1. Perangkat Lunak (Software)

Software atau dalam bahasa Indonesia yang berarti perangkat lunak adalah instruksi langsung komputer untuk melakukan pekerjaan yang dapat ditemukan di setiap aspek kehidupan modern dari aplikasi yang kritis untuk hidup (*life-critical*), seperti perangkat pemantauan medis dan pembangkit tenaga listrik sampai perangkat hiburan, seperti *video game* (Simarmata, 2010). Menurut Roger S (2002) mendefinisikan perangkat lunak sebagai berikut, “Perintah program komputer yang bila di eksekusi memberikan fungsi dan unjuk kerja seperti yang di inginkan.”. Menurut Melwin (2007) mendefinisikan perangkat lunak sebagai berikut, “Berfungsi sebagai pengatur aktivitas kerja komputer dan semua intruksi yang mengarah pada sistem komputer. Perangkat lunak menjembatani interaksi *user* dengan komputer yang hanya memahami bahasa mesin.”.

❖ Klasifikasi Perangkat Lunak (Software)

Menurut Melwin (2007:22) secara umum perangkat lunak ini dapat diklasifikasikan menjadi 2, yaitu:

1. Perangkat Lunak Sistem Operasi

Operating system software merupakan perangkat lunak yang berfungsi untuk mengkonfigurasi komputer agar dapat menerima berbagai perintah dasar yang diberikan sebagai masukan. Perangkat lunak sistem juga merupakan sekumpulan program yang ditulis untuk melayani program-program yang lain. Contoh perangkat lunak system operasi antara lain *MS-DOS*, *LINUX*, *UNIX*, dll.

2. Perangkat Lunak Aplikasi

Aplikasi yang digunakan dalam pengolahan data baik yang berukuran kecil maupun besar bisa digunakan secara *stand-alone* (tunggal) maupun dalam sistem berbasis jaringan lokal *client server* maupun *webbase* (intranet maupun internet). Contoh perangkat lunak aplikasi antara lain *MS SQL*, *Oracle*, *MySQL*, dll.

3. *Graphic Design*

Aplikasi yang digunakan untuk membuat sketsa dua dimensi untuk desain ruangan, perabotan, mesin-mesin model industri, juga model tubuh manusia, hewan, dll. Contoh perangkat lunak aplikasi untuk graphic design antara lain *AutoCad*, *Pro Design*, *CorelDraw*, dll. Ada juga yang memasukan dalam golongan ini aplikasi *image*, *audio*, *video* manipulator, yaitu aplikasi yang digunakan untuk mengolah citra, suara hingga gambar tiga dimensi untuk pembuatan animasi, pengolahan *image*, *photo*, *audio*, *video editing*, dll. Contoh aplikasi yang dimaksud antara lain *Adobe Photoshop*, *After Effect*, *Premire*, & *Audition*, *Corel Draw*, *Corel RAVE*, *Corel Photo Paint*, *GIMP (In Linux)*, *Macromedia Flash*, *Director*, dll.

4. Antivirus dan Utility Tools

Software ini digunakan untuk operasi minor (*Background*) yang bekerja di belakang sistem, berguna untuk meningkatkan kinerja sistem operasi atau aplikasi dan juga performa hardware. Terdapat 2 jenis utilitas di dalam sistem operasi:

- a) Utilitas Internal, sudah termasuk dalam paket yang di instalasi dalam sistem operasi seperti *scandisk* di *Windows* atau perintah *fsck (file system check)* di *OS Linux*.
- b) Utilitas eksternal, yang harus diinstalasi kemudian pada OS, seperti *System Optimizer*, *PC Tools*, *Data/File Recovery*, *PC Maintenance*, *Internet Utilities*, *Norton System Work*, *Registry Clean and Repair*, *Hardware Repair*.

Sementara untuk proteksi virus dan antivirus, contohnya adalah antivirus produk *Mc Affee*, *Grisoft*, *Norton*, *PandaSoft*, dll.

5. *Development Tools*

Merupakan sistem bahasa pemrograman komputer yang dilengkapi dengan compiler tertentu sebagai sumber media penyusunan program aplikasi. Contoh perangkat lunak untuk *development tools* antara lain *Visual Basic*, *Turbo Pascal*, *Visual C++*, dll. Termasuk dalam jenis aplikasi ini adalah *web development tools*, yaitu aplikasi yang digunakan untuk membuat/mendesain tampilan *website (homepage)* di internet, seperti *FrontPage*, *Macromedi Dreamweaver*, *PHPedit*, *WebPage Editor*, dll.

6. Aplikasi Komunikasi

Aplikasi yang digunakan untuk membangun komunikasi antara satu komputer dengan komputer lain dalam jaringan komputer lokal maupun global. Contoh aplikasi komunikasi seperti *E-mail & Fax Application*, *YM*, *ICQ*, *MSN*, *Internet Telephony*, dll.

7. Jenis Aplikasi Lain

- *Multimedia Player*, seperti *WinAmp*, *XMMS*, *Power DVD*, *JetAudio*, *Windows Media Player*, dll.
- *Hacker Tools*, seperti *Find Password*, *Keygen/Crack Software License*, *Scanning Port*, dll.
- *Network & Security Application*, seperti *Firewall*, *Intruder Detections*, *Network Monitoring*, dll.
- *PC Games Application*, dan masih banyak lagi.

2.2. Sistems *Development Life Cycle* (SDLC)

2.2.1. Definisi

SDLC atau *Software Development Life Cycle* atau sering disebut juga *Sistem Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (A.S. & M.Shalahuddin, 2011). Jadi SDLC dapat diartikan pula sebagai metode yang menjadi acuan dalam proses pengembangan perangkat lunak.

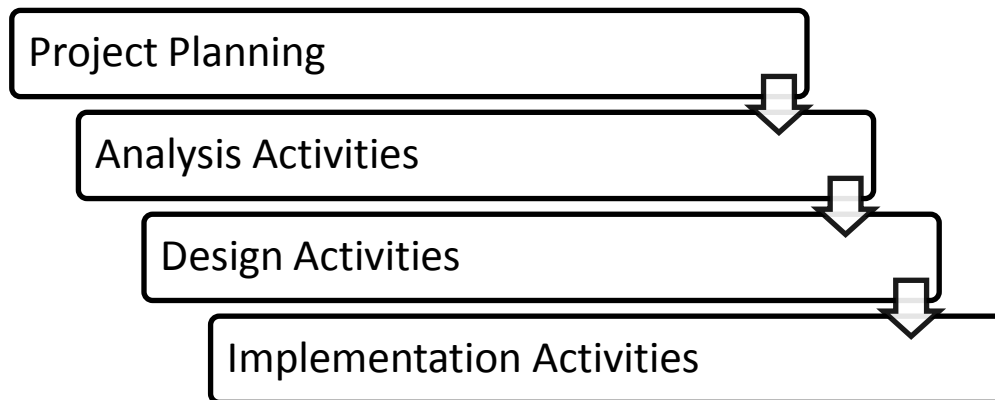
2.2.2. Tahapan dalam SDLC

SDLC sejatinya adalah langkah–langkah yang harus diikuti dalam melakukan pengembangan sistem. Berikut adalah tahapan–tahapan yang terdapat dalam SDLC secara umum (Satzinger, Jackson, & Burd, 2009):

1. *Project Planning* (Perencanaan Proyek)

Tujuan utama dari perencanaan proyek adalah melakukan identifikasi terhadap sistem baru yang akan dikerjakan, memastikan bahwa proyek ini *feasible*, membuat jadwal, melakukan perencanaan terhadap sumber, serta melakukan perhitungan *budget*. Maka tahapan dalam perencanaan proyek adalah:

- Mendefinisikan permasalahan
- Menghasilkan jadwal proyek
- Memastikan kemungkinan proyek
- Mencari orang–orang yang mengerjakan proyek
- Melaksanakan proyek



Gambar 1. Tahapan SDLC secara umum

2. *Analysis Activities* (Kegiatan Analisis)

Tujuan utama dari kegiatan analisis ialah untuk memahami dan mendokumentasikan kebutuhan bisnis dan kebutuhan proses dari sistem yang baru. Dalam kegiatan analisis, kata kunci utamanya adalah *discovery* (penemuan) dan *understanding* (pemahaman). Kegiatan yang dilakukan selama proses analisis ialah:

- Mengumpulkan informasi
- Mendefinisikan kebutuhan sistem
- Membuat *prototype* untuk kepentingan penemuan (*discovery*)
- Memprioritaskan kebutuhan
- Menghasilkan dan mengevaluasi pilihan
- Melakukan *review* terhadap rekomendasi kepada manajemen

3. *Design Activities* (Kegiatan Desain)

Tujuan utama dari kegiatan desain ialah untuk membuat solusi sistem yang nantinya akan dipakai berdasarkan kebutuhan yang telah didefinisikan dan dipilih selama kegiatan analisis. Dalam kegiatan desain terdapat dua jenis desain yaitu *High-level design* dan *low-level design*. *High-level design* berisi pengembangan struktur asitektural untuk komponen perangkat lunak, *database*, *user interface*, dan lingkungan

pengoperasian. *Low-level design* berisi tentang perincian algoritma dan struktur data yang dibutuhkan untuk pengembangan perangkat lunak. Kegiatan yang dilakukan selama proses desain adalah:

- Mendesain dan mengintegrasikan jaringan
- Mendesain arsitektur aplikasi
- Mendesain *user interface*
- Mendesain *system interface*
- Mendesain dan mengintegrasikan *database*
- Membuat *prototype* untuk mendapatkan detail dari desain
- Mendesain dan mengintegrasikan *control system*

4. *Implementation Activities* (Kegiatan Implementasi)

Kegiatan implementasi ialah kegiatan akhir dimana sistem telah dibuat, telah dilakukan pengetesan, dan telah dipasang. Tujuan dari kegiatan implementasi tidak semata-mata untuk menghasilkan sistem informasi yang *reliable* dan berfungsi secara penuh, namun juga untuk memastikan bahwa semua pekerja telah dilatih dan perusahaan siap menerima keuntungan yang diharapkan setelah menggunakan sistem yang baru. Kegiatan yang berlangsung selama proses implementasi ini ialah:

- Membangun komponen perangkat lunak
- Melakukan verifikasi
- Mengubah data
- Melatih para pengguna dan mendokumentasikan sistem
- Memasang system

2.3. **Model SDLC Waterfall**

Ada banyak metodologi yang dapat dipelajari dan digunakan dalam pengembangan perangkat lunak. Karena banyaknya metodologi yang dipelajari dan digunakan maka tidak jarang kita menemukan masalah dalam mengembangkan

perangkat lunak. Metodologi adalah sekumpulan aturan, aktivitas, dan prosedur yang digunakan dalam aktifitas penelitian. Penelitian disini digantikan dengan kata pengembangan/rekayasa perangkat lunak.

Dalam melakukan pengembangan perangkat lunak seringkali ditemukan permasalahan seperti (Hibbs, Jewett, & Sullivan, 2009):

- Proyek melebihi jadwal yang ditentukan.
- Penggunaan *budget* melebihi rencana.
- Tidak menemukan kebutuhan dari klien.
- Proyek dibatalkan.

Dari keempat hal ini dapat disimpulkan bahwa tidak semua usaha pengembangan perangkat lunak berjalan dengan semestinya. Pada tahun 1994 *Standish Group* mencanangkan *CHAOS study* yang mempelajari tentang usaha pengembangan perangkat lunak. *Standish Group* telah mempelajari sekitar 8000 proyek pengembangan perangkat lunak. *CHAOS study* telah meneliti mulai dari perusahaan kecil, sedang, hingga yang besar meliputi: perbankan, *securities*, *manufacturing*, *retail*, *wholesale*, pelayanan kesehatan, asuransi, hingga pemerintahan pusat (jumlah seluruhnya sekitar 365 perusahaan). Dari semua proses pengembangan perangkat lunak yang terjadi di perusahaan – perusahaan tersebut yang berhasil hanyalah sekitar 16% saja. Hal ini membuktikan bahwa 84% dari proyek pengembangan perangkat lunak terjadi kegagalan atau mengalami permasalahan yang sangat berat (Hibbs, Jewett, & Sullivan, 2009). Pada tahun 2004, sepuluh tahun dari penelitian sebelumnya, *Standish Group* melanjutkan penelitian mereka dan kali ini jumlah proyek perangkat lunak yang mereka amati mencapai sekitar 40000 proyek. Tingkat keberhasilan proyek – proyek tersebut pun meningkat menjadi 29%, namun tetap saja hasil ini bukanlah hasil yang bisa dibanggakan (Hibbs, Jewett, & Sullivan, 2009).

CHAOS *study* juga menggunakan wawancara terhadap perorangan ataupun kelompok untuk mendapatkan konteks kualitatif sebagai hasil dari *survey* yang dilakukan. CHAOS *study* juga membagi proyek kedalam tiga kategori, yaitu (Hibbs, Jewett, & Sullivan, 2009):

- Proyek Berhasil (*Project Succeeded*)
Proyek selesai dengan waktu dan *budget* yang sesuai dengan perencanaan dan juga sistem yang dibangun sesuai dengan yang dispesifikasikan.
- Proyek Bermasalah (*Project Challenged*)
Proyek selesai, namun melebihi batas waktu bahkan melebihi penggunaan *budget* yang ditentukan dan juga fitur yang ditawarkan di dalam sistem lebih sedikit daripada yang telah dispesifikasikan.
- Proyek Dibatalkan (*Project Canceled*)
Proyek dibatalkan pada poin tertentu di dalam tahapan pengembangan perangkat lunak (*study* ini menyebutkan proyek ini sebagai *project impaired*)

Dalam pengembangan perangkat lunak kegagalan ini bisa terjadi karena penggunaan metodologi yang sangat beresiko. Salah satu metodologi pengembangan perangkat lunak yang memiliki resiko paling tinggi menemui kegagalan adalah metode *waterfall*. Model metodologi *waterfall* mengisyaratkan setiap tahapannya menjadi sekuensial seperti tahapan air yang jatuh (air terjun) (Giachetti, 2010). Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan “*classic life cycle*”. Model ini pertama kali yang diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, Model ini juga biasa digunakan pada proyek yang menyangkut pemerintahan dan juga perusahaan – perusahaan yang sudah besar (Munassara & Govardhan, 2010). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Haerani & dkk, 2012). Walaupun model ini sudah sangat lama, model ini masih dipelajari hingga saat ini. Universitas Islam Indonesia Yogyakarta adalah salah

satu universitas swasta di Indonesia yang masih mengajarkan model *waterfall* dalam mata kuliah Rekayasa Perangkat Lunak di jurusan Teknik Informatika mereka. Teduh Dirgahayu selaku Lektor Kepala di Fakultas Teknologi Industri, Jurusan Teknik Informatika menyatakan “model *waterfall* menjadi fokus pembelajaran kami di mata kuliah rekayasa perangkat lunak, tetapi juga dikenalkan model pengembangan perangkat lunak yang lainnya”. Beliau juga menyatakan kalau model *waterfall* ini masih sangat umum karena proses yang disampaikan sangat jelas dan berurutan, sehingga sangat bagus untuk dipelajari mahasiswa yang masih pemula dalam dunia perangkat lunak. Sehingga secara tidak langsung model *waterfall* ini juga melatih proses berfikir *programmer* untuk bisa berfikir secara sistematis. Tahapan dalam model *waterfall* dapat dilihat pada Gambar 2 (Doom, 2009):

- *Requirements* (Kebutuhan)

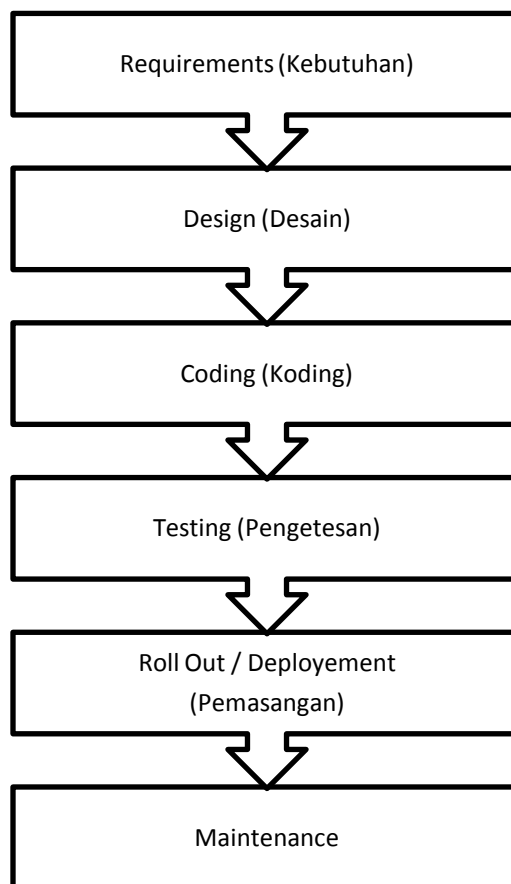
Tahap pengumpulan kebutuhan untuk aplikasi yang akan dibangun. Tahap ini dapat dikerjakan dengan melakukan penelitian terhadap model aplikasi dan model proses bisnis yang nantinya akan dibantu menggunakan aplikasi yang dibangun. Pada tahap ini lebih memfokuskan kepada “apa” yang akan dilakukan oleh aplikasi.

- *Design* (Desain)

Tahap ini berisi tentang proses menspesifikasikan bagaimana nantinya saat aplikasi mengimplementasikan kebutuhan. Desain aplikasi biasanya terdiri dari dua kegiatan yaitu:

- a. *Basic Design* (desain dasar). Pada tahap ini, kebutuhan dianalisis lebih detail dan terperinci. Contoh, aplikasi dibagi dan dibuat menjadi komponen yang terpisah.

- b. *Detailed Design* atau *Technical Design* (desain yang detail). Pada tahap ini difokuskan kepada *internal logic* dari komponen aplikasi yang terpisah dan melihat bagaimana komponen-komponen yang terpisah tersebut bekerja sama. Tahapan ini juga memuat deskripsi lengkap bagaimana sistemnya akan bekerja. Hal itu juga termasuk pertimbangan terkait aspek teknis dari aplikasi, seperti manajemen masalah penyimpanan, masalah *database*, penggunaan secara simultan dari banyak *user*, menangani *error*, dst.



Gambar 2. Model SDLC Waterfall

- *Coding* (Koding)

Berawal dari *detailed design*, *programmer* memulai membuat *code* dari aplikasi. Dan juga, *code* dari setiap komponen selalu diuji agar dapat melihat apakah aplikasi bekerja sesuai dengan keinginan. Jika terjadi *error* (“*bugs*”) maka

programmer harus segera memperbaikinya dan melakukan pengecekan ulang hingga tidak ada lagi kesalahan. Proses ini disebut pula dengan *unit testing*, karena setiap komponen (*unit*) dilakukan pengujian secara terpisah.

- *Testing* (Pengujian)

Setelah semua komponen telah diperiksa, maka hal yang selanjutnya dilakukan adalah melakukan pengujian pada aplikasi secara keseluruhan. Pengujian fungsi dari aplikasi ini disebut juga *functional test*. Dan juga, dilakukan pengujian terhadap efisiensi dari penggunaan banyak *user* serta pengujian terhadap volume data yang dibutuhkan, pengujian seperti ini disebut pula *performance test*. Setiap terjadi kesalahan maka *programmer* harus segera memperbaikinya, dan proses pengujian diulang mulai dari *unit test*.

- *Roll Out / Deployment* (Penyebaran)

Pada tahap ini aplikasi telah siap untuk digunakan. Aplikasi dipasang pada sistem operasional di mana sistem tersebut akan digunakan. Pada tahap ini juga para pengguna akan diberi pelatihan. Pada tahap ini dikenal istilah *conversion*, yaitu proses transisi dari sistem yang lama menjadi sistem yang baru. Dalam proses *conversion*, data dipindahkan dari sistem lama ke sistem yang baru dan dimungkinkan bahwa pengguna dapat menggunakan sistem yang lama dan yang baru secara paralel.

- *Maintenance* (Pemeliharaan)

Aplikasi sebaiknya dipelihara dan dijaga selama sistem masih digunakan. Dan juga, pengguna sebaiknya meminta tambahan atau merubah fungsional yang ada pada sistem dengan yang baru. Jika seperti ini maka model *waterfall* akan berjalan lagi hingga proses *roll out*.

Dari gambar di atas dapat dilihat bahwa tidak ada proses yang bisa kembali ke proses sebelumnya. Banyak pengembang perangkat lunak yang tidak percaya bahwa metode *waterfall* dapat diterapkan dalam banyak proyek. Seperti dalam makalah yang berjudul “*A Comparison Between Five Models Of Software Engineering*” menyatakan bahwa hal yang membuat model ini tidak banyak diterapkan dalam

berbagai pengembangan perangkat lunak adalah model ini sangat kaku. Satu tahapan dapat dijalankan asalkan tahapan sebelumnya telah selesai, tidak ada tahapan yang mengalami *overlap* dalam model SDLC *waterfall*. Padahal dalam kehidupan nyata pasti akan ditemukan masalah pada setiap tahapan dikarenakan kurangnya informasi yang didapatkan pada tahap analisis kebutuhan (Munassara & Govardhan, 2010). Hal ini menyebabkan, setiap tahapan diperlukan *review* dan dokumen tertentu yang dimana hal ini menyebabkan waktu tambahan pada pengembangan perangkat lunak.

Dari hal ini dapat dilihat bahwa tidak semua kegiatan pengembangan perangkat lunak dapat dilakukan dengan model SDLC *waterfall*. Ada beberapa kategori pengembangan perangkat lunak yang dapat dilakukan dengan model SDLC *waterfall*. Kategori pengembangan perangkat lunak sebagai berikut (tutorialspoint, 2013):

- Kebutuhan didokumentasi dengan sangat baik dan jelas.
- Definisi produk sudah jelas.
- Tidak ada kebutuhan yang masih ambigu.
- Sumber untuk pengerjaan produk sangat banyak dan dibutuhkan ahli untuk mendukung pengerjaan produk.

2.3.1. Keuntungan Model Waterfall

Keuntungan yang bisa didapat oleh model *waterfall* adalah (tutorialspoint, 2013):

- Mudah dipahami dan diimplementasikan.
- Sudah banyak dikenal dan diketahui.
- Membangun karakter pengembang yang baik: define before design (definisi sebelum desain), design before code (desain sebelum kode).
- Dapat menghasilkan dokumen yang terinci.
- Bekerja sangat baik untuk produk yang matang namun dengan tim yang biasa.

2.3.2. Kerugian Model *Waterfall*

Model *waterfall* adalah salah satu model yang disebut sebagai “*crystalline*” manajemen proyek. Rencana dalam model *waterfall* terkenal sangat kaku dan pembelajaran kelompok juga tidak termasuk di dalam sistemnya. Hal ini dikarenakan tidak dapat kembalinya proses jika proses selanjutnya sudah dimulai dan juga setiap proses memiliki independensi masing–masing sehingga setiap anggota hanya memiliki tanggung jawab masing–masing, hal ini lah yang menyebabkan model *waterfall* sebagai model yang memiliki resiko tinggi (Pries & Quigley, 2011). Selain itu kerugian dalam menggunakan model SDLC *waterfall* adalah:

- Tidak realistis, karena membutuhkan analisis kebutuhan yang akurat pada awal proyek.
- Penemuan yang lambat akan menghambat proses menghasilkan produk dan akan mengakibatkan kesalahan yang serius.
- Sulit untuk diintegrasikan dengan manajemen resiko.
- Sulit dan akan memakan banyak dana serta waktu untuk membuat dokumentasi.

BAB III

METODOLOGI PENELITIAN

Penelitian ini bertujuan untuk mengetahui bagaimana penggunaan model SDLC *waterfall* pada perusahaan PT. Rabiha Pilar Informatika dalam usaha pengembangan perangkat lunak. Meski, model SDLC *waterfall* ini memiliki tingkat resiko yang tinggi dalam penggunaannya (Pries & Quigley, 2011). Dalam praktik model ini masih digunakan oleh beberapa pengembang perangkat lunak, karena model ini termasuk model yang sangat mudah dipahami oleh tim pengembangan perangkat lunak terutama pengembang perangkat lunak yang masih tergolong baru (Pries & Quigley, 2011). Penelitian dilakukan dengan melihat secara langsung bagaimana para pengembang perangkat lunak di PT. Rabiha Pilar Informatika mengimplementasikan model *waterfall* ini dalam proyek pengembangan.

3.1. Penelitian Kualitatif

3.1.1. Definisi dan Alasan Penggunaan Metode Kualitatif

Sebagai sebuah metode penelitian, pendekatan kualitatif memiliki banyak nama dalam beberapa disiplin ilmu. Antropologi menamakan etnografi; sosiologi menyebutkan *verstehen* atau pengamatan terlibat; psikologi menamakan *folklore*. linguistik, etnomuskologi, etnometodologi, dan banyak disiplin ilmu lainnya, menggunakan istilah seperti studi kasus, *interpretative inquiry*, *natural inquiry*, dan *phenomenology* (Patilima, 2007).

Tugas akhir ini menggunakan metode penelitian kualitatif untuk menyelesaikan permasalahan yang ada. Pendekatan kualitatif merupakan pendekatan yang humanistik, karena di dalam pendekatan ini cara pandang, cara hidup, selera, ataupun ungkapan emosi dan keyakinan dari warga masyarakat

yang diteliti sesuai dengan masalah yang diteliti, juga termasuk data yang harus dikumpulkan seperti yang dipaparkan oleh Parsudi Suparlan, antropolog dari Universitas Indonesia. Sedangkan John W. Creswell (1994) mendefinisikan pendekatan kualitatif sebagai sebuah proses penyelidikan untuk memahami masalah sosial atau masalah manusia, berdasarkan pada penciptaan gambar holistik yang dibentuk kata – kata, melaporkan pandangan informan secara terperinci, dan disusun dalam sebuah latar ilmiah. Creswell (2008) juga mendefinisikan metode penelitian kualitatif adalah suatu pendekatan atau penelusuran untuk mengeksplorasi dan memahami suatu gejala sentral.

Pendekatan kualitatif berkaitan erat dengan sifat unik pada realitas sosial dan dunia tingkah laku manusia itu sendiri. Sasaran penelitian ini adalah kehidupan sosial dan atau masyarakat sebagai sebuah satuan atau sebuah kesatuan yang menyeluruh. Prof Parsudi Parlan (1994) menyatakan bahwa biasanya pendekatan ini dikaitkan dengan pendekatan holistik di mana pendekatan tersebut tidak mengambil sampel, melainkan melakukan penelitian secara mendalam, teliti dan menyeluruh untuk memperoleh gambaran mengenai prinsip-prinsip umum atau pola-pola yang berlaku umum sehubungan dengan gejala-gejala yang ada di dalam kehidupan sosial masyarakat yang diteliti sebagai kasus itu sendiri (Patilima, 2007).

Menurut Merriam yang dikutip oleh John W. Creswell, ada enam asumsi dalam pendekatan kualitatif yang perlu diperhatikan oleh peneliti, yaitu:

- Peneliti kualitatif lebih menekankan perhatian pada proses, bukan pada hasil atau produk
- Peneliti kualitatif tertarik pada makna, bagaimana orang membuat hidup, pengalaman, dan struktur kehidupannya masuk akal
- Peneliti kualitatif merupakan instrumen pokok untuk pengumpulan dan analisis data. Data didekati melalui instrumen manusia, bukan melalui inventaris, daftar pertanyaan atau alat lain

- Peneliti kualitatif melibatkan kerja lapangan. Peneliti secara fisik berhubungan dengan orang, latar belakang, lokasi atau institusi untuk mengamati atau mencatat perilaku dalam latar alamiahnya
- Peneliti kualitatif bersifat deskriptif dalam arti peneliti tertarik proses, makna, dan pemahaman yang didapat melalui kata atau gambar
- Peneliti kualitatif bersifat induktif, peneliti membangun abstrak, konsep, proposisi, dan teori.

Sehingga penelitian kualitatif dianggap tepat dalam menyelesaikan tugas akhir ini berdasarkan poin–poin di atas.

3.2. Subjek Penelitian

3.2.1. Teknik Pemilihan Subjek

Dalam penelitian ini pemilihan subjek dilakukan dengan menggunakan teknik *purposive sampling*. Teknik ini berdasarkan pada ciri–ciri atau sifat–sifat yang diperkirakan mempunyai sangkut paut erat dengan ciri–ciri dan sifat–sifat yang ada di dalam populasi yang sudah diketahui sebelumnya (Narbuko & Achmadi, 2003).

3.2.2. Jumlah Subjek Penelitian

Dalam tugas akhir ini jumlah sampel yang akan diwawancarai adalah dua personil dari tim pengembangan perangkat lunak yang ada di PT. Rabiha Pilar Informatika. Personil tim tersebut memiliki peran dan *job description* yang berbeda beda. Divisi yang menjadi informan dari wawancara tugas akhir ini adalah:

- Project Manajer (Manajer Proyek)
- Analisis

Dalam penelitian kualitatif tidak memiliki pedoman khusus dalam menentukan jumlah sampel. Biasanya jumlah sampel cenderung sedikit karena tujuan dari penelitian kualitatif adalah mendapatkan informasi secara mendalam. Beberapa teks riset merekomendasikan enam hingga delapan unit data, jika sampelnya berupa unit homogen, atau, dua belas hingga dua puluh unit data, jika sampelnya berupa unit heterogen (Kuzel, 1999). Beberapa riset yang sangat mendalam didasarkan pada sampel yang sangat kecil (terutama untuk riset fenologi), sebab hal ini memungkinkan peneliti untuk berfokus pada kedalaman suatu fenomena, bukannya pada isu–isu permukaan bercakupan luas (Daymon & Holloway, 2008).

Penelitian kualitatif cenderung menggunakan sampel yang sedikit ketimbang penelitian kuantitatif. Sehingga penggunaan kata sampel dalam penelitian kualitatif kurang tepat. Sebagai gantinya, jumlah sampel yang diteliti cukup dikatakan jumlah partisipan yang terlibat (Istijanto, 2009). Menurut Patton (2006), faktor kunci dalam menyeleksi dan membuat keputusan tentang unit analisis yang tepat adalah memutuskan unit apakah yang peneliti inginkan bisa dapat mengatakan sesuatu tentang akhir dari evaluasi.

3.3. Metode Pengumpulan Data

Penelitian ini menggunakan metode wawancara sebagai metode pengumpulan datanya. Dengan wawancara peneliti dapat menggali tidak saja apa saja yang diketahui dan dialami subjek yang diteliti, akan tetapi apa yang tersembunyi di dalam diri subjek penelitian. Selain itu, apa yang ditanyakan kepada informan bisa mencakup hal–hal yang bersifat lintas waktu, yang berkaitan dengan masa lampau, masa sekarang, dan juga masa yang akan mendatang.

Wawancara yang digunakan adalah wawancara kualitatif. Dimana peneliti mengajukan pertanyaan–pertanyaan secara bebas dan leluasa, tanpa terikat oleh suatu susunan pertanyaan yang telah dipersiapkan sebelumnya. Namun tentu saja peneliti juga mempersiapkan *interview guide* yang perlu ditanyakan kepada informan.

Interview guide disusun berdasarkan teori yang ada dan *interview guide* dibuat dengan maksud agar wawancara bisa lebih terfokus dan dapat mengungkapkan data lebih dalam (Sugiyono, 2012). Berikut *interview guide* dalam penelitian ini:

- Project Planning (Perencanaan Proyek)
 - Alasan menggunakan *waterfall*
 - Lama pengerjaan menggunakan *waterfall*
 - Tahapan pengerjaan
 - Penentuan *resource* (Sumber daya manusia dan *budget*)
- Analisis Kebutuhan
 - Jumlah SDM dan *budget*
 - Lama pengerjaan
 - Metode analisis
 - Masalah yang dihadapi
- Perancangan
 - Jumlah SDM dan *budget*
 - Lama pengerjaan
- *Development* (Pengembangan)
 - Yang berperan dalam pengembangan
 - Jumlah SDM dan *budget*
 - Lama pengerjaan
 - Masalah yang dihadapi
- *Testing*
 - Jumlah SDM dan *budget*
 - Lama pengerjaan
 - Metode *testing*
- *Release*
 - Proses *release*
- *Maintenance*

- Proses *maintenance*

3.4. Alat Bantu Pengumpulan Data

Dalam penelitian ini peneliti menggunakan wawancara sebagai metode pengumpulan data dan menggunakan alat bantu berupa alat perekam suara atau gambar dan alat tulis berupa *pc tablet* serta menggunakan peralatan menulis.

3.5. Prosedur Penelitian

3.5.1. Keterbatasan Penelitian

Penelitian ini dilakukan dengan menggunakan metode kualitatif dengan data primer yang diperoleh melalui wawancara. Keterbatasan pada penelitian ini disebabkan oleh subyektifitas yang ada pada peneliti. Penelitian ini sangat bergantung kepada interpretasi peneliti tentang makna yang tersirat dalam wawancara sehingga kecenderungan untuk bias masih tetap ada. Untuk mengurangi bias maka dilakukan proses triangulasi. Proses triangulasi merupakan proses yang umum digunakan untuk melakukan pengujian validitas pada penelitian kualitatif. Proses triangulasi yang dilakukan adalah triangulasi sumber data dan metode. Menurut Patton triangulasi sumber data menggunakan berbagai sumber data seperti dokumen, arsip, hasil wawancara, hasil observasi atau juga dengan mewawancarai lebih dari satu subjek yang dianggap memiliki sudut pandang yang berbeda. Dengan kata lain triangulasi sumber data dilakukan cross check data dengan fakta dari informan yang berbeda. Triangulasi metode adalah penggunaan berbagai metode untuk meneliti suatu hal, dalam penelitian ini peneliti menggunakan metode wawancara untuk mendapatkan hasil penelitian.

3.5.2. Tahap Persiapan

Peneliti melakukan persiapan dengan melakukan pencarian dan identifikasi masalah yang ditemukan dalam pengembangan perangkat lunak.

Kemudian peneliti melakukan tinjauan kepustakaan dengan mengumpulkan informasi dari internet dan literatur yang terkait dengan topik yang akan dibahas.. Setelah mendapatkan beragam informasi, peneliti melakukan seleksi terhadap informasi yang didapatkan dan dirangkum untuk keperluan penelitian. Setelah itu, peneliti menentukan metode penelitian serta menentukan teknik pengambilan data dan menentukan informan yang akan diwawancarai.

3.5.3. Tahap Pelaksanaan

Pada tahap ini dilakukan pengambilan data melalui teknik wawancara. Wawancara dilakukan dengan memberikan keleluasaan waktu kepada informan, karena dalam wawancara kualitatif, wawancara bisa memakan waktu yang cukup lama.

Peneliti memulai proses pengambilan data dengan melakukan pengenalan terlebih dahulu serta menjelaskan tujuan dari penelitian yang dilakukan. Selanjutnya peneliti melakukan wawancara sesuai dengan *interview guide* yang telah dibuat. Dalam proses wawancara pertanyaan akan mengalami perkembangan seraya mengikuti jawaban yang diberikan oleh informan demi mendapatkan jawaban yang lebih mendalam. Peneliti juga melakukan observasi terhadap respon yang berupa mimik wajah dan gerak tubuh informan yang muncul ketika informan menjawab pertanyaan atau menceritakan pengalaman mereka. Setelah wawancara selesai, peneliti mengucapkan terima kasih serta akan menanyakan kesediaan informan untuk diwawancarai lagi apabila peneliti membutuhkan informasi kembali. Data mengenai waktu dan lokasi pengambilan ditunjukkan pada tabel 1.

No	Nama Informan	Jabatan	Waktu Pengambilan Dara	Lokasi Pengambilan Data
1.	Hari Setiaji	<i>Team Leader</i>	Rabu, 23 Juli 2013, pukul 15.00 – 16.16	PT. Rabiha Pilar Informatika
2.	Chanifah Indah Ratnasari	Analisis	Rabu, 23 Juli 2013, pukul 15.00 – 16.16	PT. Rabiha Pilar Informatika

Tabel 1. Tabel Pengambilan Data

3.5.4. Tahap Analisis Data

Pada penelitian kualitatif, data yang didapatkan berupa data kualitatif. Data kualitatif adalah data yang diperoleh dari hasil pengumpulan data dan informasi dengan menggunakan berbagai metode pengumpulan data, seperti pengamatan, wawancara, menggambar, disuksi kelompok terfokus, dan lain-lain (Patilima, 2007). Semua data tersebut diterima dan dianalisis melalui tahapan–tahapan kegiatan analisis data kualitatif. Proses pengumpulan data pada analisis data kualitatif dibagi menjadi beberapa tahapan, yaitu (Lacey & Luff, 2001):

1) Transkripsi

Tahap ini berupa kegiatan mentransfer hasil rekaman wawancara yang dilakukan ke dalam bentuk format penulisan. Namun dalam kegiatan ini biasanya peneliti cenderung bias, karena data yang dimasukkan hanya berupa data yang relevan terhadap penelitian, padahal data bisa saja berupa gerak gerak (non verbal) informan selama proses wawancara.

2) Pengorganisasian Data

Tahap ini berupa kegiatan mencatat keterangan informan seperti nama perusahaan informan, pengalaman informan dalam bidang pengembangan perangkat lunak, yang mana informasi itu akan digunakan dalam tahap penyajian data.

3) Pengenalan

Tahap ini berupa proses mendengarkan atau menonton ulang kegiatan wawancara yang telah dilakukan, serta membuat memo dan rangkuman serta membaca kembali data yang didapatkan.

3.5.5. Tahap Penyajian Data

Prinsip dasar penyajian data adalah membagi pemahaman kita tentang sesuatu hal yang pada orang lain. Oleh karena itu data yang diperoleh adalah kata-kata dan bukan bentuk angka, sering kali data adalah sebuah kutipan langsung dari kata-kata informan. Hasil penelitian kualitatif juga dapat disajikan dalam bentuk *life History*, yaitu deskripsi tentang peristiwa dan pengalaman penting dari kehidupan atau beberapa bagian pokok dari kehidupan seseorang dengan kata-katanya sendiri.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Karakteristik Informan

Informan yang digunakan dalam penelitian ini adalah anggota dari tim pengembang perangkat lunak yang berada pada perusahaan PT. Rabiha Pilar Informatika. Dalam penelitian ini menggunakan dua subjek penelitian yaitu:

1. Hari Setiaji, *team leader* dari tim pengembang perangkat lunak di PT. Rabiha Pilar Informatika
2. Chanifah Indah Ratnasari, analis dari tim pengembang perangkat lunak di PT. Rabiha Pilar Informatika

PT. Rabiha Pilar Informatika adalah salah satu perusahaan yang bergerak dalam bidang pengembangan perangkat lunak di Kota Yogyakarta. Perusahaan ini telah berdiri sejak tahun 2006 hingga saat ini. Produk utama dari perusahaan ini pada awalnya adalah sistem informasi manajemen pendidikan, namun perusahaan ini telah beralih menjadi sistem informasi manajemen rumah sakit. Perusahaan ini juga menerima beberapa pesanan khusus yang berkenaan dengan sistem informasi.

4.2. Hasil Wawancara

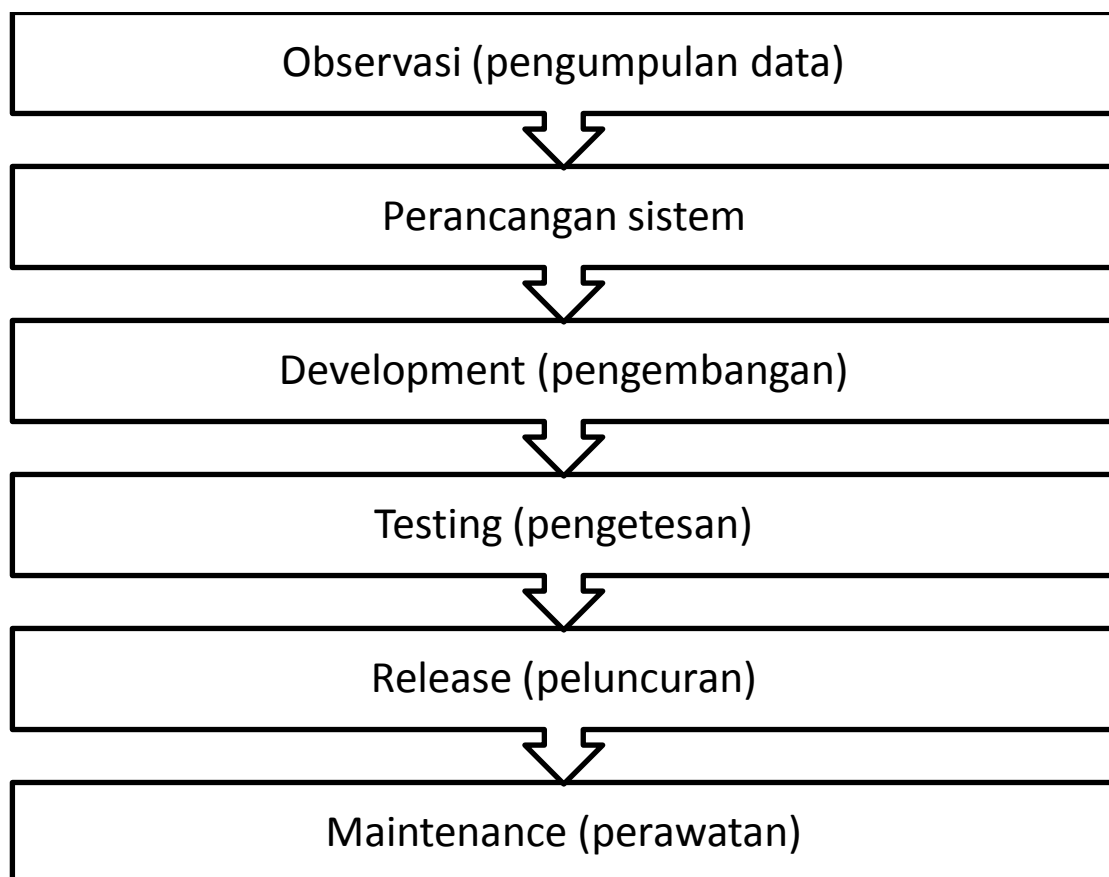
Hasil wawancara dalam penelitian ini dimaksudkan untuk merangkum dan memilih hal-hal yang dibutuhkan atau memfokuskan hal-hal yang penting dari pengambilan data melalui wawancara.

4.2.1. Model *Waterfall* yang Digunakan

Model SDLC *waterfall* adalah model tertua yang dari semua model SDLC yang ada. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini

sering disebut dengan “*classic life cycle*”. Model ini pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno. Model ini juga biasa digunakan pada proyek yang menyangkut pemerintahan dan perusahaan-perusahaan yang sudah besar (Munassara & Govardhan, 2010).

Pada umumnya model *waterfall* memiliki alur tahapan proses analisis kebutuhan, desain, coding, *testing*, pemasangan, perawatan. PT. Rabiha Pilar Informatika adalah perusahaan yang menggunakan model *waterfall* untuk mengerjakan proyek pengembangan perangkat lunak dalam perusahaan mereka. Model *waterfall* yang mereka gunakan ditunjukkan pada gambar 3.



Gambar 3. Alur Proses Model *Waterfall* PT. RPI

Model *waterfall* yang diterapkan pada PT. Rabiha Pilar Informatika dimulai dari tahap pengumpulan data atau yang disebut dengan analisis kebutuhan dimana peran klien sangat dibutuhkan dalam tahap ini. Pada tahap ini pengembang dan klien bertemu untuk membicarakan tentang sistem yang akan digunakan oleh klien. Klien memberikan penjelasan berupa kata-kata maupun dalam bentuk dokumen. Hasil dari analisis kebutuhan tersebut akan dilanjutkan ke tahap perancangan sistem. Berangkat dari hasil analisis yang dilakukan perancangan sistem mengikuti apa yang telah dituliskan pada analisis kebutuhan.

Setelah tahapan perancangan sistem (desain) telah dilakukan maka akan dilanjutkan ke tahap pengembangan. Di sini peran *developer* sangat penting, namun dalam tahap ini analisis PT. Rabiha Pilar Informatika masih berperan sebagai penghubung antara klien dengan *developer*. Setelah sistem selesai maka akan dilanjutkan dengan kegiatan pengetesan (*testing*). Dalam tahap ini pengetesan dilakukan oleh *tester internal*. Jika *tester* telah selesai melakukan pengetesan terhadap sistem yang telah dibuat maka sistem tersebut sudah layak untuk diluncurkan (*release*). Setelah peluncuran, PT. Rabiha Pilar Informatika juga melakukan perawatan terhadap sistem yang mereka hasilkan.

4.2.2. Tahapan Model *Waterfall* PT. Rabiha Pilar Informatika

Berdasarkan dari penjelasan sebelumnya, PT. Rabiha Pilar Informatika adalah perusahaan yang menggunakan model *waterfall* untuk menyelesaikan proyek yang mereka kerjakan. Tahapan model *waterfall* pada perusahaan tersebut adalah:

- Observasi/Pengumpulan data (analisis kebutuhan)
- Perancangan Sistem
- *Development* (pengembangan)
- Testing (pengetesan)
- Release (peluncuran)
- Maintenance (perawatan)

4.2.2.1. Project Planning (Perencanaan Proyek)

Dalam tahapan SDLC (*Software/system development life cycle*) sejatinya dimulai dari tahapan analisis kebutuhan. Dalam model *waterfall* tahapan ini adalah tahapan yang sangat penting, karena dalam tahap ini peran klien sangat dibutuhkan, terutama dalam hal penyedia informasi untuk pembangunan sistem yang mereka minta. Peran klien menjadi sangat penting karena dalam tahap model *waterfall*, klien tidak akan berinteraksi lagi sampai sistem telah selesai. Maka jika terjadi kesalahan atau informasi yang di dapat dalam tahapan kurang, maka bisa diperkirakan bahwa sistem yang dibuat akan mengalami kegagalan.

PT. Rabiha Pilar Informatika menggunakan model *waterfall* untuk beberapa kondisi pengerjaan perangkat lunak. Perusahaan ini menggunakan model ini dikarenakan pengetahuan dan pengalaman tentang sistem yang akan dibuat sedikit. Sesuai dengan pernyataan Hari terkait alasan penggunaan model *waterfall* di PT. Rabiha Pilar Informatika.

Kami menggunakan waterfall karena knowledge dan pengalaman kami untuk membuat sistem yang direquest masih sangat kecil. Dalam waterfall juga perancangan sangat ditekankan, maka dari itu kami menggunakan model waterfall (Hari).

Jadi dikarenakan kurangnya pengetahuan tentang sistem yang dibuat maka perusahaan ini menggunakan model *waterfall*. Dalam *waterfall*, tahap analisis sangatlah penting, karena jika terjadi kesalahan atau kurangnya informasi yang di dapat bisa berakibat fatal terhadap sistem yang dibuat.

Penggunaan *waterfall* juga dipengaruhi oleh klien yang meminta. Perusahaan ini mengakui bahwa untuk menggunakan *waterfall*, biasanya klien yang membutuhkan sistem informasi adalah klien dari perusahaan besar.

Kami menggunakan model waterfall biasanya karena kliennya dari korporasi atau instansi besar yang meminta lingkup proyek yang besar (Hari).

Dikarenakan lingkup proyek yang besar, maka proses analisisnya pun harus memakan waktu yang lama untuk mengurangi kesalahan sistem yang akan dibuat. Hari juga menjelaskan bagaimana suatu proyek dikatakan memiliki lingkup proyek yang besar.

Proyek yang besar ini bisa dibilang memiliki kompleksitas sistem yang beragam, estimasi waktu pengerjaan yang lama, dan kebutuhan resource (SDM dan biaya) yang besar (Hari).

Dalam penggunaan model *waterfall* biasanya memakan waktu yang lama dalam pengerjaannya. Seperti yang dinyatakan oleh Hari terkait dengan masalah waktu dalam pengerjaan proyek menggunakan model *waterfall*.

Dalam menggunakan model waterfall biasanya kami bisa memakan waktu hingga satu tahun lebih (Hari).

Untuk waktu pengerjaan proyek mulai dari proses analisis kebutuhan hingga waktu peluncuran, biasanya PT. Rabiha Pilar Informatika bisa memakan waktu hingga satu tahun bahkan lebih.

Proyek yang menggunakan model *waterfall* membutuhkan *budget* yang cukup besar. Seperti yang dinyatakan oleh Hari terkait dengan *budget* yang mereka gunakan selama proses pengembangan perangkat lunak menggunakan model *waterfall*.

Kami bisa menghabiskan budget mulai dari 100 hingga 150 juta selama menjalani proyek yang menggunakan model waterfall. Budget yang kami habiskan untuk gaji perorangan tim, kebutuhan eksternal sistem, dan bonus (Hari).

Untuk kebutuhan eksternal berdasarkan keterangan Hari, biasanya berupa:

- Kebutuhan hardware
- Kebutuhan software
- Desain
- Database
- Server

nilai dari bonus yang dinyatakan oleh Hari bisa mencapai sepuluh persen dari nilai proyek yang mereka kerjakan. Jadi berdasarkan keterangan tersebut maka *budget* yang dibutuhkan selama mengerjakan proyek menggunakan model *waterfall* memang tergolong besar.

4.2.2.2. Analisis Kebutuhan dan Perancangan Sistem

Analisis kebutuhan adalah tahapan awal dari semua tahapan yang ada pada SDLC. Tahapan ini berisi tentang interaksi antara pengembang dan klien untuk membicarakan/mendiskusikan tentang sistem yang akan dibuat. Dalam *waterfall* analisis kebutuhan menjadi hal yang sangat penting. Model *waterfall* tidak mengijinkan adanya proses yang kembali ke proses sebelumnya jika proses selanjutnya telah berjalan. Maka dari itu, jika proses analisis mengalami kegagalan atau kurangnya informasi yang didapat bisa berakibat kegagalan dalam sistem yang dibuat. Dalam model *waterfall*, interaksi dengan klien hanya terjadi saat tahap analisis dan implementasi. PT. Rabiha Pilar Informatika hanya

menggunakan satu atau dua orang sebagai analis untuk menjalankan tahapan ini.

Dalam tahap analisis atau observasi kami menggunakan satu atau dua orang sebagai analis dan analis inilah yang akan berinteraksi dengan klien untuk mendiskusikan sistem yang akan dibuat. Dari pihak klienpun sudah ditentukan siapa yang akan menjadi juru bicara untuk mendiskusikan masalah tersebut (Hari).

Hal yang dibicarakan antara pengembang dan klien berupa proses bisnis yang akan diimplementasikan ke dalam sistem yang akan dibuat. Informasi yang didapatkan bisa berupa keterangan dari klien maupun dokumen resmi yang dibuat oleh klien.

Tahapan analisis kebutuhan adalah tahapan yang paling lama dalam model *waterfall*. PT. Rabiha Pilar Informatika juga mengakui bahwa untuk menjalani tahap analisis ini mereka membutuhkan waktu yang cukup lama.

Ideal pengerjaan tahap analisis kebutuhan biasanya satu hingga tiga bulan, tapi terkadang jika mengalami masalah yang cukup rumit bisa mencapai enam bulan. Jadi bisa dikatakan bahwa tahap analisis bisa memakan waktu hingga 50% waktu pengerjaan proyek (Hari).

Tahapan analisis sendiri memakan *budget* yang cukup banyak. Dalam tahapan ini *budget* yang dikeluarkan selain gaji dari analis bisa juga berupa akomodasi untuk bertemu dengan klien.

Dalam tahapan analisis biasanya biaya yang dikeluarkan cukup banyak terutama jika kliennya jauh. Budget untuk analis sendiri bisa mencapai empat jutaan per bulan untuk satu orang, sedangkan untuk akomodasi menyesuaikan dengan transport yang digunakan (Hari).

Media analisis kebutuhan yang digunakan oleh PT. Rabiha Pilar Informatika yaitu bisa bertemu langsung dengan klien maupun bisa via *chatting*. Seperti yang dinyatakan oleh Hari mengenai media analisis kebutuhan yang mereka jalani.

*Biasanya kami dalam melakukan analisis kebutuhan menyesuaikan jarak antara klien dengan kami. Jika kliennya masih dalam satu kota maka bisa langsung bertemu, namun jika kliennya jauh kami bisa melalui via *chatting* atau bisa bertemu secara langsung dengan menyesuaikan budget yang ada untuk akomodasi (Hari)*

Untuk melakukan pertanyaan PT. Rabiha Pilar Informatika menggunakan metode *TOP-DOWN* dan *BOTTOM-UP* untuk menggali informasi yang diberikan oleh klien.

Pas analisis kami menggunakan metode TOP-DOWN dan BOTTOM-UP untuk menanyakan informasi terkait dengan proses bisnis yang yang berimplikasi terhadap sistema yang akan dibuat. Maksudnya TOP-DOWN adalah menanyakan alur proses bisnis mulai dari awal hingga akhir yang nantinya akan diimplementasikan ke dalam sistem. Dan BOTTOM-UP adalah menanyakan secara detail tentang bagaimana proses bisnis tersebut berjalan (Hari).

Setelah melalui tahapan analisis kebutuhan, maka akan selanjutnya akan dilakukan tahap perancangan sistem. Pada perusahaan PT. Rabiha Pilar Informatika Perancangan sistem masih dikerjakan oleh analis dari tahapan analisis kebutuhan. Output yang diharapkan dari tahapan ini adalah dokumen perancangan yang nantinya akan digunakan oleh *developer* untuk membuat sistem yang diminta. Berikut adalah pernyataan Hari mengenai isi dari dokumen perancangan sistem.

Isi dari perancangan sistem nanti berupa rancangan proses bisnis yang sudah diimplementasikan ke dalam sistem, rancangan antar muka, dan rancangan database yang akan digunakan (Hari).

Dalam tahapan perancangan PT. Rabiha Pilar Informatika terkadang tidak melakukan perancangan secara *internal*, melainkan perusahaan ini membeli rancangan serupa yang cocok untuk digunakan dalam sistem yang mereka kerjakan.

Tidak semua rancangan sistem, kami yang mengerjakan, tapi kami juga terkadang membeli dari pihak lain untuk memudahkan pekerjaan kami (Hari).

Biaya untuk pembelian desain tersebut telah dimasukkan diawal penentuan *budget* pada tahap perencanaan proyek.

4.2.2.3. Development (Pengembangan)

Development atau pengembangan adalah tahapan setelah analisis kebutuhan dan perancangan sistem. *Input* dari tahapan ini adalah dokumen perancangan sistem yang dikerjakan oleh analis dalam perusahaan PT. Rabiha Pilar Informatika. Dalam tahapan ini yang memiliki peranan penting adalah *developer/programmer* yang bertugas untuk membangun sistem yang diminta dan dibantu oleh analis sebagai pengawas selama pengerjaan sistem.

Pada tahap development, yang memiliki peranan paling penting adalah developer atau programmer, namun dalam tahap ini analis bertindak sebagai support jikalau ada permasalahan dalam dokumen perancangan sistem (Hari).

PT. Rabiha Pilar Informatika menggunakan tiga orang *programmer/developer* dalam tahapan ini.

Kami hanya menggunakan tiga orang yang bertindak sebagai developer atau programmer. Kami membagi menjadi dua jenis developer pertama developer senior dan junior. Biasanya senior hanya satu orang yang bertindak mengawasi dua junior (Hari).

Menurut Hari, waktu ideal untuk tahap pengembangan pada model *waterfall* di PT. Rabiha Pilar Informatika adalah minimal tiga bulan.

Untuk masa pengembangan, biasanya dalam perusahaan kami mematok minimal tiga bulan, namun terkadang bisa juga molor jika terdapat permasalahan dalam dokumen perancangan (Hari).

Dalam masa pengembangan, PT. Rabiha Pilar Informatika hanya mengeluarkan biaya untuk gaji individu *developer*.

Di development kami biasanya hanya mengeluarkan budget untuk gaji para developer, namun gaji tersebut disesuaikan dengan tingkatan developer. Untuk developer yang senior berkisar antara dua hingga dua setengah juta per bulan dan untuk junior biasanya berkisar antara satu setengah hingga dua juta per bulan (Hari)

Setelah masa *development* selesai, para *developer* melakukan tes masing-masing sebelum diserahkan pada *tester* untuk tahapan pengetesan

4.2.2.4. Testing (Pengetesan)

Setelah masa *development* selesai maka tahapan selanjutnya adalah tahapan pengetesan. Pada PT. Rabiha Pilar Informatika, tahapan ini berisi kegiatan pengetesan sistem secara keseluruhan yang dilakukan oleh tester yang berjumlah satu orang.

Pada tahap testing kami menggunakan satu orang sebagai tester, dan testernya pun juga dari internal (Hari)

Model pengetesan yang dilakukan pada PT. Rabiha Pilar Informatika berupa pengetesan kepada fitur-fitur yang telah selesai dibuat oleh developer.

Testing yang kami lakukan biasanya testing per fitur. Jadi fitur-fitur yang sudah selesai bisa langsung ditesting oleh tester. Indikator bahwa fitur tersebut sudah layak untuk digunakan sudah tersimpan di analisis kebutuhan, jadi para tester cukup melakukan cek apakah fitur tersebut sudah sesuai dengan analisis kebutuhan yang dibuat (Hari)

Contoh gambar fitur yang sedang dilakukan pengetesan:

Halaman Login	
Username	:
Password	:
submit	
Description	:
halaman ini adalah halaman login user	
Kondisi Input	:
user bisa memasukkan username user bisa memasukkan password tombol submit bisa ditekan untuk masuk ke halaman selanjutnya	
Kondisi Output	:
masuk ke halaman awal dengan priviledge user	
Review	:

Gambar 4. Halaman Testing

Menurut Hari, waktu yang paling ideal dalam masa testing di PT. Rabiha Pilar Informatika adalah dalam waktu dua hingga tiga hari, karena

testing juga dilakukan per fitur bukan per modul, jadi masa untuk masa testing bisa menjadi lebih cepat.

Untuk masa testing biasanya kami mematok waktu dua hingga tiga hari. Karena testing yang kami lakukan disini testing per fitur, jadi pengerjaannya juga bisa lebih cepat (Hari)

PT. Rabiha Pilar Informatika juga mengahbiskan budget untuk testing hanya berupa gaji para tester, tidak ada pengeluaran lain dalam tahap ini.

Pengeluaran pada masa testing hanya berupa gaji individu tester, gajinya ada pada range satu hingga satu setengah juta per bulan (Hari)

4.2.2.5. Release (Peluncuran)

Setelah tahapan testing selesai dan semua fitur dinyatakan lolos, PT. Rabiha Pilar Informatika meluncurkan sistem tersebut ke server yang mereka miliki yaitu server “Rabiha”.

Habis testing, dan dinyatakan lolos oleh tester, maka langkah selanjutnya kami upload sistem tersebut ke dalam server kami yaitu Rabiha. Jadi sistem bekerja secara online dan kami juga bisa memantau jika terjadi masalah (Hari).

Dalam masa testing biasanya PT. Rabiha Pilar Informatika memakan waktu hingga enam bulan bahkan lebih.

Karena kamu meluncurkan sistem per fitur, maka biasanya untuk meluncurkan satu sistem penuh bisa memakan waktu hingga enam bulan bahkan lebih (Hari).

4.2.2.6. Maintenance (Perawatan)

Masa *maintenance*/perawatan adalah masa dimana sistem yang telah dibuat dilakukan pengecekan dan klien dapat meminta bantuan dalam perbaikan jika terjadi masalah kecil dalam sistem yang PT. Rabiha Pilar Informatika kerjakan. PT. Rabiha Pilar Informatika berani menawarkan jasa *maintenance life-time* selama dalam perawatan tidak menuntut untuk mengubah sistem secara drastis.

Untuk hal maintenance kami berani memberika pelayanan life-time, karena sistem yang kami buat juga terintegrasi dengan sistem yang ada pada perusahaan kami, jadi kami dapat dengan mudah mengontrol jika terjadi masalah pada klien kami. Namun kami tidak melayani jika sistem tersebut minta diubah, kalau begitu kami ingin hal tersebut dijadikan proyek yang baru (Hari)

4.2.3. Masalah Dalam Penggunaan Model Waterfall

Masalah-masalah yang dihadapi oleh PT. Rabiha Pilar Informatika dalam menerapkan model SDLC waterfall adalah sebagai berikut:

Klien :

- Informasi yang akan diberikan pada pengembang belum siap

- Alur proses bisnis yang masih belum jelas
- Komunikasi dengan pengembang kurang baik
- Klien masih bingung apa yang mereka inginkan

Tim :

- Pengetahuan akan proses bisnis masih kurang
- Komunikasi dengan klien masih kurang baik
- Suasana hati dan pikiran pegawai yang tidak menentu
- Pengetahuan *programmer junior* masih belum memadai
- Pengerjaan dokumen yang begitu banyak

Timeline (Jadwal) :

- Jadwal mundur dari waktu yang dijanjikan
- Jam kerja sering bertambah

Budget (Dana) :

- Dana pengerjaan kurang memadai atau pas tanpa adanya keuntungan yang diterima

4.3. Tabel Rangkuman Penelitian

Tahapan Proses	Tema	Sub Tema
Project Planning	Alasan menggunakan <i>waterfall</i>	Kompleksitas sistem yang besar Dibutuhkan dana yang

		besar Pengetahuan akan sistem yang akan dibuat masih kurang
	Lama pengerjaan menggunakan <i>waterfall</i>	≥ 1 tahun
	Tahapan pengerjaan	Observasi (analisis kebutuhan) Perancangan sistem Development Testing Maintenance
	Penentuan <i>resource</i> (Sumber daya manusia dan <i>budget</i>)	Resource SDM menggunakan 5 – 6 orang dalam 1 tim Penghitungan budget : Gaji perorangan, kebutuhan eksternal, bonus (10% nilai proyek)
Observasi (Analisis Kebutuhan) & Perancangan sistem	Jumlah SDM dan <i>budget</i>	Jumlah SDM 1 – 2 orang Budget $\geq$ RP. 4 juta/bulan/orang
	Lama pengerjaan	1 – 6 bulan
	Metode analisis	TOP-DOWN dan BOTTOM-UP
	Masalah yang dihadapi	Data dari klien belum siap Proses bisnis klien yang

		belum pasti Komunikasi belum baik Suasana hati pegawai Waktu pengerjaan yang lama Analisis belum memahami proses bisnis klien
Development	Yang berperan dalam pengembangan	Analisis sebagai pengawas Developer/programmer sebagai pembuat sistem
	Jumlah SDM dan <i>budget</i>	3 orang (1 senior dan 2 junior)
	Lama pengerjaan	≥ 3 bulan
	Masalah yang dihadapi	Kesulitan mengikuti permintaan Suasana hati pegawai
Testing	Jumlah SDM dan <i>budget</i>	1 orang
	Lama pengerjaan	2 – 3 hari (per fitur)
	Metode <i>testing</i>	Testing dilakukan setiap ada fitur yang selesai
Release	Proses release	Setiap fitur yang telah selesai dilakukan pengetesan akan langsung di release ke server PT. Rabiha Pilar Informatika Lama proses release

		hingga seluruh sistem jadi ≥ 6 bulan
Maintenance	Proses maintenance	Karena sistem yang telah dikerjakan sudah terintegrasi dengan server PT. Rabiha Pilar Informatika, maka maintenance bisa dilakukan secara jarak jauh dari perusahaan PT. Rabiha Pilar Informatika berani memberikan jasa perawatan <i>life-time</i> selama tidak mengubah sistem yang ada

Tabel 2. Tabel Tema dan Sub Tema Hasil Wawancara

4.4. Pembahasan

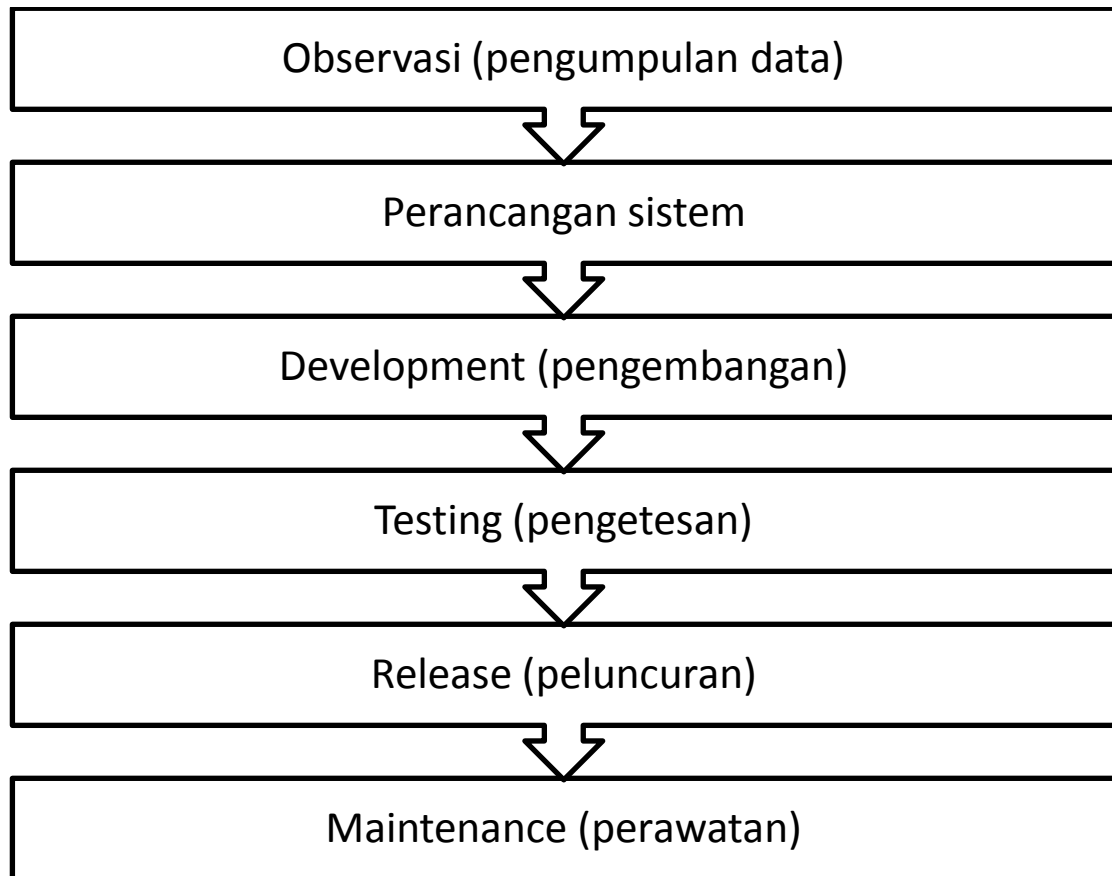
Pembahasan dalam penelitian ini akan mengulas beberapa data yang didapatkan dalam hasil wawancara dan beberapa dokumen yang dirangkum oleh peneliti dari buku maupun berita-berita di internet. Ulasan ini berisikan data penelitian yang meliputi tahapan proses model waterfall yang digunakan oleh perusahaan PT. Rabiha Pilar Informatika, alasan menggunakan model waterfall, banyak tenaga yang dibutuhkan setiap prosesnya, budget yang dibutuhkan untuk mengerjakan sebuah proyek menggunakan model waterfall, serta proses release dan maintenance yang terjadi di perusahaan tersebut.

Salah satu perusahaan pengembangan perangkat lunak yang ada di Kota Yogyakarta adalah PT. Rabiha Pilar Informatika. Perusahaan ini telah menyetujui untuk dimintai keterangan mengenai proses pengembangan perangkat lunak menggunakan model SDLC *waterfall*. Perusahaan ini bergerak di bidang pengembangan sistem informasi rumah sakit. Pada awalnya perusahaan ini bergerak di bidang pengembangan sistem informasi pendidikan. Namun terkadang perusahaan ini menerima beberapa permintaan dalam pengembangan sistem informasi yang lain. Baru-baru ini perusahaan tersebut menerima pekerjaan untuk membuat sistem informasi untuk salah satu perusahaan media cetak di Kota Yogyakarta. Namun dokumen untuk proyek ini tidak dapat dilihat untuk dijadikan bahan penelitian kali ini.

Perusahaan ini telah berdiri sejak tahun 2006 hingga sampai saat ini. Hari Setiaji adalah team leader dari perusahaan ini. Hari juga menjadi narasumber untuk penelitian ini untuk mendapatkan informasi terkait penggunaan model waterfall untuk menyelesaikan proyek yang sedang dikerjakan. Hari mengakui masih menggunakan model waterfall untuk beberapa kondisi proyek. Menurut Hari, proyek yang akan dikerjakan jika ingin menggunakan model waterfall adalah proyek yang memiliki kompleksitas yang tinggi, membutuhkan waktu yang lama dalam proses analisisnya, serta membutuhkan *resource* (SDM dan budget) yang cukup banyak. Hal ini juga dibenarkan oleh salah satu makalah “*A Comparison Between Five Models Of Software Engineering*” disitu dinyatakan bahwa model waterfall tidak dapat dikerjakan dalam semua proyek, karena model waterfall tidak mengijinkan adanya proses yang berjalan “naik” ke proses sebelumnya, jika proses selanjutnya sudah berjalan (Munassara & Govardhan, 2010). Maka dalam model ini dibutuhkanlah analisis yang sangat matang, sehingga kemungkinan terjadi kesalahan pada pengembangan perangkat lunak bisa dikurangi. Pada model waterfall, klien dan pengembang hanya berinteraksi pada dua tahap yaitu pada tahap analisis dan tahap implementasi, hal ini juga dibenarkan oleh Indah, analis dari perusahaan PT. Rabiha Pilar Informatika. Indah membenarkan pernyataan tersebut selaku analis dalam

proyek pengembangan perangkat lunak. Maka dari itu, proses analisis biasanya memakan waktu paling lama dalam menggunakan model waterfall.

Tahapan model waterfall yang digunakan oleh PT. Rabiha Pilar Informatika dapat dilihat pada gambar 5.



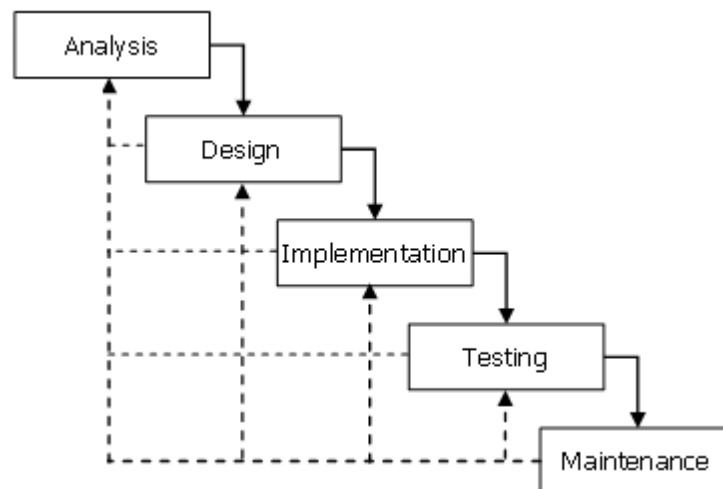
Gambar 5. Model Waterfall PT. RPI

Menurut Indah dan Hari alur proses model waterfall yang mereka terapkan adalah mulai dari tahap pengumpulan data atau yang disebut dengan analisis kebutuhan dimana peran klien sangat dibutuhkan dalam tahap ini. Pada tahap ini pengembang dan klien bertemu untuk membicarakan tentang sistem yang akan digunakan oleh klien. Klien memberikan penjelasan berupa kata-kata maupun dalam bentuk dokumen. Hasil dari analisis kebutuhan tersebut akan dilanjutkan ke tahap

perancangan sistem. Berangkat dari hasil analisis yang dilakukan perancangan sistem mengikuti apa yang telah dituliskan pada analisis kebutuhan.

Setelah tahapan perancangan sistem (desain) telah dilakukan maka akan dilanjutkan ke tahap pengembangan. Di sini peran *developer* sangat penting, namun dalam tahap ini analis PT. Rabiha Pilar Informatika masih berperan sebagai penghubung antara klien dengan *developer*. Setelah sistem selesai maka akan dilanjutkan dengan kegiatan pengetesan (*testing*). Dalam tahap ini pengetesan dilakukan oleh *tester internal*. Jika *tester* telah selesai melakukan pengetesan terhadap sistem yang telah dibuat maka sistem tersebut sudah layak untuk diluncurkan (*release*). Setelah peluncuran, PT. Rabiha Pilar Informatika juga melakukan perawatan terhadap sistem yang mereka hasilkan.

Model ini adalah model yang paling sederhana seperti yang ada pada makalah “*A Simulation Model for the Waterfall Software Development Life Cycle*”, disitu dinyatakan bahwa model waterfall yang paling umum adalah (2012):



Gambar 6. Model Waterfall

Dalam makalah “*A Comparison Between Five Models Of Software Engineering*” dinyatakan bahwa ada dua model waterfall yang bisa digunakan, yaitu model *pure*

waterfall dan *modified waterfall*. Perbedaan dari dua model ini adalah *modified waterfall* lebih fleksibel dibandingkan dengan model *pure waterfall* (Munassara & Govardhan, 2010). Dalam praktiknya, PT. Rabiha Pilar Informatika menggunakan model *modified waterfall*, karena seperti yang dinyatakan oleh Hari, mereka membagi sistem yang besar menjadi beberapa modul, dan setiap modul dapat dijalankan secara parallel namun tetap dengan mengikuti pedoman model *waterfall*.

PT. Rabiha Pilar Informatika (RPI) menggunakan lima sampai enam orang dalam pengembangan perangkat lunak menggunakan model *waterfall*. Menurut Hari, setiap proyek yang menggunakan *waterfall* pasti menghabiskan dana dan waktu yang cukup lama. PT. RPI bisa menghabiskan waktu lebih satu tahun dan dana lebih dari seratus juta untuk sekali melakukan pengembangan perangkat lunak menggunakan model *waterfall*. Dana tersebut terbagi atas:

- Gaji SDM (Sumber Daya Manusia)
- Kebutuhan eksternal klien seperti hardware dan software
- Bonus pegawai (10% nilai proyek)

Tahapan pertama dari model *waterfall* PT. RPI adalah observasi/analisis kebutuhan. Pekerjaan ini dilakukan oleh satu orang dari tim. Dalam tahap ini analis akan bertemu dengan perwakilan dari klien untuk mendiskusikan kebutuhan dari sistem yang akan dikerjakan. PT. RPI biasanya bertemu langsung dengan klien atau melalui *chatting* untuk mendapatkan informasi yang diperlukan. Informasi tersebut bisa berupa kata-kata, dokumen, atau *history* dari *chatting*. Metode yang digunakan oleh PT. RPI untuk mendapatkan informasi yang diperlukan adalah *TOP-DOWN* dan *BOTTOM-UP*. *TOP-DOWN* adalah menanyakan alur proses bisnis mulai dari awal hingga akhir yang nantinya akan diimplementasikan ke dalam sistem dan *BOTTOM-UP* adalah menanyakan secara detail tentang bagaimana proses bisnis tersebut berjalan.

Setelah mendapatkan informasi yang dibutuhkan, analis segera membuat dokumen perancangan sistem, yang nantinya akan digunakan oleh developer untuk mengimplementasikan sistem yang dibuat. Perancangan sistem PT. RPI berisi tentang pembuatan rancangan proses bisnis yang diimplementasi ke dalam sistem, antarmuka, dan struktur database. Namun terkadang proses desain sistem tidak dilakukan oleh PT. RPI, melainkan mereka membeli desain yang cocok untuk sistem yang mereka bangun. Dana pembelian desain ini telah dimasukkan ke dalam pengeluaran dana eksternal. Kegiatan analisis dan perancangan sistem di PT. RPI biasanya memakan waktu satu hingga enam bulan. Dan memakan biaya berupa gaji pegawai sekitar empat juta. Jika klien berada pada jarak yang jauh maka biaya akomodasi telah direncanakan sejak awal termasuk biaya eksternal.

Masalah-masalah yang ditemukan selama proses analisis dan perancangan di PT. RPI adalah:

- Data dari klien belum siap
- Alur proses bisnis klien belum pasti
- Komunikasi belum terjalin dengan baik
- Analis tidak mengerti proses bisnis klien
- Suasana hati pegawai

Dari masalah-masalah ini akan berakibat pada:

- Lamanya waktu analisis
- Dana yang berlebihan
- Waktu kerja yang berlebih

Dan jika klien belum memahami apa yang menjadi keinginan mereka, PT. RPI biasanya memberikan masukan berupa gambaran *output* dari sistem yang akan mereka kerjakan, hal ini juga merupakan solusi untuk mengurangi kemungkinan “molornya” waktu pengerjaan.

Setelah tahap analisis dan perancangan sistem telah selesai, maka tahap selanjutnya adalah tahap *development*/pengembangan/implementasi. Tahap ini berisi tentang kegiatan pembuatan kode untuk sistem yang akan dibangun. Pembangunan sistem berangkat dari hasil analisis dan rancangan yang telah dibuat. Di tahap ini, PT. RPI masih menggunakan peran analis. Pada tahap ini analis bertugas sebagai pengawas. Analis juga bertindak sebagai *support* jika ada *programmer* yang tidak memahai isi dari rancangan sistem. Di PT. RPI, mereka menggunakan tiga developer dalam satu proyek. Biasanya terdiri dari satu *programmer senior* dan dua *programmer junior*. Dalam tahap ini PT. RPI menghabiskan dana untuk membayar gaji pegawai. Terdapat perbedaan antara gaji senior dan junior, senior biasanya menerima dua hingga dua setengah juta per orang. Junior menerima satu setengah hingga dua juta per orang. Tidak ada kendala khusus untuk masalah *development*, menurut Hari masalah umum yang biasa dihadapi oleh programmer adalah tidak mengerti rancangan sistem dan suasana hati pegawai. Menurut Hari dan Indah, waktu yang ideal untuk pengerjaan tahap ini adalah minimal tiga bulan.

Masa *development* telah selesai, maka tahap berikutnya adalah *testing*/pengetesan. Dalam tahap ini dilakukannya pengetesan terhadap apa yang telah dibuat oleh developer. Di PT. RPI, Hari menyatakan bahwa pengetesan yang dilakukan oleh perusahaan mereka adalah berupa pengetesan berdasarkan fitur apa yang telah dibuat. PT. RPI selalu membagi sistem yang besar menjadi beberapa modul dan setiap modul akan dikerjakan per fitur, maka pengetesan yang dilakukan juga mengikuti fitur apa yang telah dikerjakan. Pengetesan dilakukan oleh satu orang tester dan indikator pengujiannya adalah analisis kebutuhan yang akan disesuaikan dengan fitur yang telah dikerjakan. Dalam tahapan ini, PT. RPI menghabiskan dana untuk gaji pegawai sebesar satu hingga satu setengah per orang. Dan memakan waktu untuk pengujian per fitur selama dua hingga tiga hari.

Tahap terakhir setelah pengujian dan fitur dinyatakan lulus adalah tahap peluncuran/*release*. Pada tahap ini fitur akan diunggah ke server PT. RPI yaitu

Rabiha. Dimana sistem akan mudah dikontrol untuk masalah perawatan nantinya. Menurut Hari masa release hingga sistem bisa dipakai secara keseluruhan memakan waktu selama kurang lebih enam bulan. Dan PT. RPI memberikan jasa perawatan sistem selama *life-time* selama tidak mengubah sistem yang ada.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Model *waterfall* adalah salah satu model yang digunakan sebagai acuan dalam pengembangan perangkat lunak. Model ini dikenal sebagai model yang paling tua dan kaku. Meskipun begitu, model ini masih dipelajari dan bahkan digunakan dalam PT. Rabiha Pilar Informatika Yogyakarta. Dari penelitian ini dapat disimpulkan bahwa:

1. Model ini masih digunakan untuk menyelesaikan proyek pengembangan perangkat lunak
2. PT. Rabiha Pilar Informatika tidak menggunakan model *pure waterfall* melainkan model *modified waterfall*
3. Model ini cocok digunakan untuk proyek yang membutuhkan waktu dan *resource* yang banyak

5.2. Saran

Saran bagi penelitian selanjutnya adalah sebaiknya perusahaan yang menjadi informan bagi penelitian ditambah sehingga terlihat model SDLC apa yang paling banyak digunakan dan dapat terlihat tingkat kecocokan sebuah model SDLC terhadap sebuah proyek.

DAFTAR PUSTAKA

- A.S., R., & M.Shalahuddin. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula.
- Andreas. (2007, November 11). *Model Software Development*. Retrieved April 12, 2013, from Wordpress: <http://tonyjustinus.wordpress.com/category/model-software-development/>
- Daymon, C., & Holloway, I. (2008). *Metode - Metode Riset Kualitatif dalam Public Relation*. Yogyakarta: Bentang Pustaka.
- Doom, C. (2009). *An Introduction To Business Information Management*. Brussels: ASP nv.
- Giachetti, R. E. (2010). *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton: Taylor & Francis Group, LLC.
- Haerani, D., & dkk. (2012). *Makalah Analisis Perancangan Sistem Informasi Tentang Model Waterfall*. Garut.
- Hibbs, C., Jewett, S., & Sullivan, M. (2009). *The Art of Lean Software Development*. Sebastopol: O'Reilly Media, Inc.
- Istijanto. (2009). *Aplikasi Praktis Riset Pemasaran : Cara Praktis Meneliti Konsumen dan Pesaing (Edisi Revisi)* . Jakarta: Gramedia Pustaka.
- Munassara, N. M., & Govardhan, A. (2010). *A Comparison Between Five Models Of Software Engineering* .
- Narbuko, D. C., & Achmadi, D. H. (2003). *Metodologi Penelitian*. Jakarta: Bumi Aksara.

- Nayak, R., & Qiu, T. (2006). Use of Data Mining in System *Development* Life Cycle. In G. J. Williams, & S. J. Simoff, *Data Mining Theory, Methodology, Techniques, and Applications* (p. 105). Germany: Springer-Verlag Berlin Heidelberg.
- Nugroho, A. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Yogyakarta: C.V ANDI OFFSET (ANDI).
- Patilima, H. (2007). *Metode Penelitian Kualitatif*. Bandung: Alfabeta.
- Petersen, K., Wohlin, C., & Baca, D. (2009). The *Waterfall* Model in Large-Scale.
- Pries, K. H., & Quigley, J. M. (2011). *Scrum Project Management*. Boca Raton: Taylor & Francis Group.
- S.Balaji, & Murugaiyan, D. (2012). *WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC* .
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2009). *System Analysis and Design in a Changing World, Fifth Edition*. Boston: Course Technology.
- Schwalbe, K. (2009). *Information Technology Project Management 6E*. United States of America: Course Technology.
- Simarmata, J. (2010). *Reakyasa Perangkat Lunak*. Yogyakarta: ANDI Offset.
- tutorialspoint. (2013). *SDLC Waterfall*. Retrieved Mei 30, 2013, from tutorialspoint.com: http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- Umi Proboyekti, S. M. (2012). Bahan Ajar Rekaya Perangkat Lunak. *SOFTWARE PROCESS MODEL I* .

LAMPIRAN



UNIVERSITAS ISLAM INDONESIA
Jurusan Teknik Informatika FTI

SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Radix Megantoro

No. Mhs. : _____

Judul TA : _____

Coba cari apakah ada metode ilmiah & mengukur efektivitas
model Waterfall !

Nilai kemajuan Tugas Akhir: _____ (0 - 100)
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 21/5/2013 .

Dosen,

(nama terang)

Ardhanyah Anfin S.T., M.Cs.

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran



UNIVERSITAS ISLAM INDONESIA
Jurusan Teknik Informatika FTI

SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Radix Megantoro

No. Mhs. : _____

Judul TA : _____

- Dibutuhkan dokumentasi formal untuk mendukung sudut pandang narasumber.
 - Referensi ilmiah ttg waterfall ?
- Tetap semangat.. :)

Nilai kemajuan Tugas Akhir: _____ (0 - 100)
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 21 Mei 2013

Dosen,


Hari Setiaji (nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran