

BAB 4 Memahami Kebutuhan Rekayasa

Perangkat Lunak

Bagi rekayasawan perangkat lunak memahami kebutuhan – kebutuhan untuk sesuatu permasalahan mungkin merupakan pekerjaan yang sulit yang biasa dihadapi. Apakah pelanggan tahu apa yang mereka butuh ? Apakah peran pengguna akhir memiliki pemahaman yang cukup baik tentang fitur – fitur dan fungsi – fungsi yang akan memberikan pada mereka keuntungan – keuntungan berkaitan dengan penggunaan perangkat lunak yang dikembangkan ? jawaban dari pertanyaan tersebut adalah tidak, walaupun para pengguna akhir dapat merumuskan kebutuhan – kebutuhan secara eksplisit namun seringkali kebutuhan – kebutuhan mereka mengalami perubahan di sepanjang waktu berjalannya proyek.

4.1. Rekayasa Kebutuhan

Sejumlah besar pekerjaan dan teknik yang dapat dilakukan untuk lebih memahami kebutuhan – kebutuhan biasanya dinamakan sebagai rekayasa kebutuhan. Dari sudut pandang proses perangkat lunak, rekayasa kebutuhan adalah tindakan – tindakan utama dalam kebanyakan rekayasa perangkat lunak yang bermula pada aktivitas – aktivitas komunikasi dan kemudian berlanjut ke aktivitas – aktivitas pemodelan . Rekayasa kebutuhan harus diadaptasi ke kebutuhan – kebutuhan proses , proyek , produk dan orang – orang yang akan mengerjakannya. Rekayasa kebutuhan membangun jembatan dari tahap analisis ke tahapan – tahapan perencanaan dan konstruksi.

Rekayasa kebutuhan pada dasarnya menyediakan mekanisme – mekanisme yang digunakan untuk memahami kebutuhan pelanggan, menyediakan mekanisme yang dapat digunakan untuk melakukan analisis kebutuhan, mekanisme – mekanisme untuk menegosiasikan solusi atas permasalahan yang memiliki alasan yang kuat, mekanisme – mekanisme untuk menspesifikasikan solusi permasalahan sehingga tidak menimbulkan makna ganda, mekanisme – mekanisme yang digunakan untuk menvalidasi spesifikasi – spesifikasi kebutuhan dan mekanisme – mekanisme yang cocok untuk mengelola kebutuhan – kebutuhan saat ditransformasikan ke dalam perangkat lunak yang bersifat operasional. Pada dasarnya rekayasa kebutuhan mencakup 7 (tujuh) pekerjaan yaitu (Roger S. Pressman, 2012) :

- I. Pengenalan masalah (*inception*)
Proyek perangkat lunak dimulai saat kebutuhan bisnis telah teridentifikasi dengan baik, atau saat pasar baru yang bersifat potensial dikenali atau layanan – layanan ditemukan. Pada pengenalan proyek, ditetapkan pemahaman dasar tentang permasalahan, tentang siapa yang menginginkan solusi bagi permasalahan itu, tentang sifat – sifat alamiah solusi yang dikehendaki dan tentang efektivitas komunikasi awal dan kolaborasi yang akan terjadi di antara para stakeholder dan tim perangkat lunak.
2. Pengenalan lanjutan (*elicitation*)
Pada pekerjaan pengenalan lanjutan dilakukan identifikasi tentang apa sesungguhnya sasaran – sasaran dari sistem atau produk yang akan dikembangkan, bagaimana sistem atau produk yang akan

dikembangkan , bagaimana sistem atau produk yang akan dikembangkan sesuai dengan kebutuhan bisnis dan yang terakhir bagaimana penggunaan sistem atau produk yang akan dikembangkan. Christel dan kang dalam (Roger S. Pressman, 2012) mengidentifikasi sejumlah permasalahan yang dijumpai saat tahap pengenalan lanjut ini dilaksanakan yaitu :

- a. Lingkup permasalahan. Batasan – batasan dari sistem tidak jelas atau para pelanggan menentukan rincian teknis yang membingungkan alih – alih menjelaskan sasaran sistem secara keseluruhan.
- b. Permasalahan yang berkaitan dengan pemahaman. Para pelanggan tidak secara lengkap merasa pasti apa yang mereka butuhkan.
- c. Permasalahan – permasalahan yang berkaitan dengan kestabilan. Kebutuhan – kebutuhan akan mengalami perubahan – perubahan di sepanjang waktu berjalannya proyek perangkat lunak.

3. Elaborasi

Elaborasi dilakukan dengan cara membuat atau menghaluskan skenario – skenario pengguna yang pada dasarnya bermanfaat untuk mendeskripsikan bagaimana para pengguna akhir akan berinteraksi dengan sistem. Masing – masing skenario pengguna diuraikan dan ditafsirkan lebih lanjut untuk mendapatkan kelas – kelas analisis yaitu entitas – entitas ranah bisnis yang tampak dari arah para pengguna akhir. Atribut – atribut masing – masing kelas analisis didefinisikan dan layanan – layanan yang diperlukan oleh masing – masing kelas diidentifikasi dan kolaborasi – kolaborasi

antar kelas diidentifikasi dan berbagai diagram pelengkap dihasilkan

4. Negoisasi

Sebagai rekayasawan sering harus mendamaikan konflik – konflik melalui negoisasi. Para pengguna, para stakeholder lainnya harus ditanyai tentang prioritas masing – masing kebutuhan dan kemudian mereka bisa diajak untuk membahas masing – masing konflik yang terjadi menurut prioritas masing – masing kebutuhan itu. Gunakan pendekatan yang sifatnya iteratif untuk melakukan penentuan skala prioritas kebutuhan – kebutuhan , menilai biaya – biaya, risiko – risikonya masing – masing, dan menyelesaikan konflik internal, dan dengan demikian rekayaswan sistem bisa menghapuskan kebutuhan – kebutuhan yang tidak diperlukan , menggabung – gabungkan dan memodifikasi kebutuhan – kebutuhan sedemikian rupa sehingga masing – masing pihak menjadi puas.

5. Spesifikasi

Terminologi spesifikasi memiliki arti yang berbeda – beda bagi setiap orang . sebuah spesifikasi dapat berupa dokumen – dokumen tertulis, sejumlah model – model grafis, suatu model matematika formal, sejumlah skenario penggunaan perangkat lunak, sebuah protipe, atau gabungan semuanya. Beberapa menyarankan agar suatu pola yang bersifat kaku dikembangkan dan digunakan untuk melakukan spesifikasi kebutuhan – kebutuhan dengan pendapat bahwa hal itu memungkinkan kebutuhan – kebutuhan bisa dipresentasikan dengan cara yang konsisten sehingga

pada gilirannya akan menjadi lebih mudah untuk dipahami.

6. Validasi

Produk – produk kerja yang dihasilkan sebagai akibat dilakukannya proses rekayasa kebutuhan dinilai untuk menilai kualitasnya pada setiap tahap validasi . validasi kebutuhan – kebutuhan dilakukan dengan cara melakukan pemeriksaan pada spesifikasi – spesifikasi untuk memastikan bahwa semua kebutuhan perangkat lunak telah didapatkan tanpa adan kemungkinan terjadinya pemahaman ganda yang membingungkan , untuk meastikan bahwa semua tidak ketidakkonsitenan, pengabaian – pengabain dan kesalahan – kesalahan yang terdeteksi telah di perbaiki, dan untuk memastikan bahwa semua produk telah sesuai dengan standar yang telah ditetapkan sebelumnya untuk proses – proses proyek dan produk perangkat lunak sendiri. Mekanisme – mekanisme validasi atas kebutuhan – kebutuhan yang paling penting adalah tinjauan – tinjauan teknis.

7. Pengelolaan kebutuhan

Pengelolaan kebutuhan merupakan sejumlah aktivitas yang akan membantu tim proyek perangkat lunak untuk mengidentifikasi , mengendalikan dan melacak kebutuhan – kebutuhan dan melacak perubahan – perubahan pada kebutuhan – kebutuhan setiap saat di dalam siklus pengembangan sistem perangkat lunak.

Pola untuk spesifikasi kebutuhan perangkat lunak

SRS (*Software Requirement Specification*) merupakan dokumen yang harus dibuat saat deskripsi rinci dari semua aspek perangkat lunak yang akan dikembangkan harus dispesifikasikan sebelum proyek perangkat lunak itu sendiri dimulai. Adapun garis besar topiknya yang diungkapkan oleh Karl Weigers dalam (Roger S. Pressman, 2012) adalah sebagai berikut :

Daftar Isi

Sejarah Revisi

- I. Pendahuluan
 - a. Kegunaan
 - b. Konvensi – konvensi(aturan) dalam dokumen
 - c. Pembaca yang dituju
 - d. Lingkup proyek
 - e. Rujukan - rujukan
2. Deskripsi Keseluruhan
 - a. Sudut pandang produk
 - b. Fitur – fitur produk
 - c. Kelas – kelas pengguna dan karakteristik – karakterisknya
 - d. Lingkungan Operasional
 - e. Perancangan dan Implementasi Batasan – batasan
 - f. Dokumentasi pengguna
 - g. Asumsi – asumsi dan ketergantungan - ketergantungan
3. Fitur – fitur sistem
 - a. Fitur sistem -I

- b. Fitur sistem 2 (dst)
- 4. Kebutuhan – kebutuhan antarmuka eksternal
 - a. Antarmuka pengguna
 - b. Antarmuka perangkat keras
 - c. Antarmuka perangkat lunak
 - d. Antarmuka komunikasi
- 5. Kebutuhan – kebutuhan non fungsional lainnya
 - a. Kebutuhan – kebutuhan kinerja
 - b. Kebutuhan – kebutuhan keamanan
 - c. Kebutuhan – kebutuhan kualitas perangkat lunak
- 6. Kebutuhan – kebutuhan lainnya

Lampiran A : Daftar Istilah

Lampiran B : Model – model Analisis

Lampiran C : Daftar Permasalahan

4.2. Menetapkan Pekerjaan Dasar

Langkah – langkah yang diperlukan untuk meletakkan kerangka kerja yang dapat digunakan untuk memahami kebutuhan – kebutuhan perangkat lunak yaitu untuk mendapatkan langkah awal proyek perangkat lunak yang kemudian akan bergerak maju untuk mendapatkan solusi – solusi atas permasalahan yang berhasil. Berikut langkah – langkah dasar dalam menetapkan pekerjaan dasar (Roger S. Pressman, 2012):

4.2.I. Mengidentifikasi orang – orang yang berkepentingan

Sommerville dan Sawyer mendefinisikan orang – orang yang berkepentingan (stakeholder) sebagai semua orang yang mendapatkan keuntungan langsung maupun tidak langsung dari perangkat lunak yang akan dikembangkan. Masing – masing orang – orang yang berkepentingan sering memiliki pandangan yang berbeda – beda tentang sistem, mendapatkan keuntungan yang berbeda saat sistem secara berhasil dikembangkan, dan mungkin saja mendapatkan berbagai risiko yang berbeda saat upaya pengembangan sistem perangkat lunak gagal. Pada tahap pengenalan , seharusnya mendaftarkan orang – orang yang akan memiliki kontribusi atas kebutuhan – kebutuhan yang akan ditangkap (Roger S. Pressman, 2012).

4.2.2. Mengenali Berbagai Sudut Pandang yang Berbeda

Karena ada banyak orang – orang yang berkepentingan , kebutuhan perangkat lunak harus digali secara mendalam dari berbagai perspektif yang berbeda. Masing – masing stakeholder akan menyumbangkan informasi – informasi tertentu pada proses rekayasa kebutuhan . saat informasi – informasi dari berbagai perspektif dikumpulkan, kebutuhan – kebutuhan itu mungkin tidak konsisten satu dengan lainnya saat digabungkan atau memiliki konflik antar kebutuhan. Dalam hal ini seharusnya melakukan pengelompokan terhadap masing – masing informasi itu sedemikian rupa sehingga para pengambil keputusan dapat memilih suatu kumpulan kebutuhan perangkat lunak secara internal konsisten (Roger S. Pressman, 2012).

4.2.3. Bekerja melalui Kolaborasi

Para pelanggan dan para stakeholder dan rekayawan perangkat lunak harus bekerjasama atau berkolaborasi jika mereka menginginkan perangkat lunak yang akan dikembangkan hasilnya bagus. Kolaborasi tidak berarti bahwa kebutuhan – kebutuhan didefinisikan oleh kelompok. Dalam banyak kasus, para stakeholder berkolaborasi dengan cara menyediakan pandangan mereka masing – masing tentang kebutuhan – kebutuhan , tetapi jagoan proyek yang kuat mungkin akan melakukan pengambilan keputusan yang sifatnya final (Roger S. Pressman, 2012).

4.2.4. Mananyakan Pertanyaan – pertanyaan pertama

Pertanyaan – pertanyaan yang ditanyakan pada tahap pengenalan suatu proyek perangkat lunak seharusnya bersifat bebas dari konteks, dan pada dasarnya berfokus pada pelanggan serta penyandang dana dan orang – orang yang berkepentingan , serta fokus juga pada sasaran dan keuntungan proyek pengembangan perangkat lunak secara keseluruhan. Pertanyaan – pertanyaan tersebut akan membantu mengidentifikasi semua stakeholder pada perangkat lunak yang akan dikembangkan. Selain itu digunakan untuk mengidentifikasi keuntungan – keuntungan yang dapat diukur dari implementasi yang berhasil dan dapat digunakan untuk mengidentifikasi adanya alternatif – alternatif lain yang mungkin untuk melakukan kustomisasi pengembangan perangkat lunak. Pertanyaan – pertanyaan tersebut juga memungkinkan rekayasawan mendapatkan pemahaman yang lebih tentang permasalahan dan memungkinkan pelanggan menyuarakan perspektif mereka tentang solusi yang ditawarkan oleh rekayasawan kebutuhan (Roger S. Pressman, 2012).

4.3. Pengenalan Permasalahan lebih lanjut untuk Mendapatkan Kebutuhan – kebutuhan

Pengenalan lebih lanjut untuk mendapatkan kebutuhan – kebutuhan menggabungkan unsur – unsur penyelesaian masalah, elaborasi, negosiasi dan spesifikasi. Tujuannya agar menjadi bersifat kolaboratif , pendekatan berorientasi pada tim untuk memperoleh kebutuhan – kebutuhan , orang – orang

yang berkepentingan harus saling bekerja sama untuk mengidentifikasi permasalahan , mengusulkan solusi – solusi atas permasalahan – permasalahan , menegosiasikan pendekatan – pendekatan yang berbeda dan menetapkan sejumlah solusi kebutuhan yang bersifat pendahuluan. Berikut langkah – langkah untuk pengenalan masalah lebih lanjut untuk mendapatkan kebutuhan – kebutuhan (Roger S. Pressman, 2012):

4.3.1. Penentuan Kebutuhan – kebutuhan yang Bersifat Kolaboratif

Panduan untuk menentukan kebutuhan – kebutuhan yang bersifat kolaboratif adalah (Roger S. Pressman, 2012):

1. Pertemuan – pertemuan dipimpin dan diikuti oleh baik rekayasawan maupun stakeholder
2. Aturan – aturan untuk persiapan dan partisipasi ditetapkan
3. Agenda dibuat dengan betuk yang cukup formal untuk bisa memperoleh semua poin yang penting.
4. Seorang fasilitator bisa ditunjuk untuk mengendalikan pertemuan
5. Mekanisme definisi dapat digunakan

Sasaran dari kegiatan ini adalah untuk mengidentifikasi permasalahan – permasalahan , mengusulkan solusi – solusi atas permasalahan – permasalahan , menegosiasikan pendekatan – pendektan yang berbeda dan mengspesifikasikan sejumlah kebutuhan dalam

atmosfir yang kondusif untuk mencapai sasaran dari solusi perangkat lunak.

4.3.2. QFD (*Quality Function Deployment*)

QFD merupakan teknik manajemen kualitas yang bekerja dengan cara menerjemahkan kebutuhan – kebutuhan pelanggan menjadi kebutuhan – kebutuhan teknis untuk perangkat lunak. QFD mengidentifikasi 3 jenis kebutuhan yaitu (Roger S. Pressman, 2012):

- I. Kebutuhan – kebutuhan normal
Tujuan – tujuan dan sasaran – sasaran produk ditentukan saat terjadinya pertemuan – pertemuan dengan para pelanggan. Jika kebutuhan – kebutuhan itu ditemukan pada produk atau sistem , para pelanggan akan terpuaskan.
2. Kebutuhan – kebutuhan yang diharapkan
Kebutuhan – kebutuhan bersifat implisit pada produk dan mungkin saja bersifat mendasar secara teknis sehingga para pelanggan tidak dapat secara eksplisit menyatakan ketidakhadiran kebutuhan – kebutuhan yang diharapkan ini akan mengakibatkan ketidakpuasan yang mendalam.
3. Kebutuhan – kebutuhan yang menyenangkan
Kebutuhan – kebutuhan yang menyenangkan ini berada di luar apa yang diharapkan para pelanggan dan terbukti akan sangat memuaskan para pelanggan itu jika kebutuhan – kebutuhan

yang menyenangkan memang dapat dipersembahkan pada mereka.

QFD biasanya menggunakan wawancara dengan pengguna dan observasi – observasi, survei – survei dan penelitian terhadap data yang bersifat historis sebagai data mentah untuk aktivitas penangkapan kebutuhan – kebutuhan. Data itu kemudian diterjemahkan kedalam tabel – tabel kebutuhan yang dinamakan sebagai tabel suara para pelanggan yang kemudian ditinjau oleh para pelanggan dan orang – orang yang berkepentingan lainnya. Sejumlah diagram, matriks dan metode evaluasi bisa digunakan untuk mendapatkan kebutuhan – kebutuhan yang diharapkan dan bisa digunakan untuk mengupayakan kebutuhan – kebutuhan yang menyenangkan (Roger S. Pressman, 2012).

4.3.3. Skenario Penggunaan

Setelah kebutuhan – kebutuhan perangkat lunak berhasil diperoleh, visi tentang keseluruhan fungsi dan fitur sistem yang akan dikembangkan mulai jelas. Namun tetap saja masih untuk bergerak langsung pada aktivitas rekayasa perangkat lunak sampai memahami fungsi – fungsi dan fitur – fitur yang akan digunakan. Untuk menyelesaikan permasalahan ini pengembang dan para stakeholder dapat membuat sejumlah skenario yang dapat digunakan untuk mengidentifikasi sejumlah penggunaan sistem yang mungkin. Skenario – skenario ini disebut use case yang menyediakan deskripsi yang rinci tentang bagaimana sistem akan digunakan (Roger S. Pressman, 2012).

| | | |
|---|---|--|
| Use case name: | Register for Conference | UniqueID: Conf RG 003 |
| Area: | Conference Planning | |
| Actor(s): | Participant | |
| Stakeholder: | Conference Sponsor, Conference Speakers | |
| Level: | Blue | |
| Description: | Allow conference participant to register online for the conference using a secure Web site. | |
| Triggering Event: | Participant uses Conference Registration Web site, enters userID and password, and clicks the logon button. | |
| Trigger type: | <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal | |
| Steps Performed (Main Path) | | Information for Steps |
| 1. Participant logs in using the secure Web server. | | userID, Password |
| 2. Participant record is read and password is verified. | | Participant Record, userID, Password |
| 3. Participant and session information is displayed on the Registration Web page. | | Participant Record, Session Record |
| 4. Participant enters information on the Registration Web form and clicks Submit button. | | Registration Web Form |
| 5. Registration information is validated on the Web server. | | Registration Web Form |
| 6. Registration Confirmation page is displayed to confirm registration information. | | Confirmation Web Page |
| 7. Credit card is charged for registration fees. | | Secure Credit Card Web Page |
| 8. Add Registration Journal record is written. | | Confirmation Web Page |
| 9. Registration record is updated on the Registration Master. | | Confirmation Web Page, Registration Record |
| 10. Session record is updated for each selected session on the Session Master. | | Confirmation Web Page, Session Record |
| 11. Participant record is updated for the participant on the Participant Master. | | Confirmation Web Page, Participant Record |
| 12. Successful Registration Confirmation Web page is sent to the participant. | | Registration Record Confirmation Number |
| Preconditions: | Participant has already registered and has created a user account. | |
| Postconditions: | Participant has successfully registered for the conference. | |
| Assumptions: | Participant has a browser and a valid userID and password. | |
| Success Guarantee: | Participant has registered for the conference and is enrolled in all selected sessions. | |
| Minimum Guarantee: | Participant was able to logon. | |
| Requirements Met: | Allow conference participants to be able to register for the conference using a secure Web site. | |
| Outstanding Issues: | How should a rejected credit card be handled? | |
| Priority: | High | |
| Risk: | Medium | |

Gambar 4.I Contoh skenario usecase
Sumber : (Kenneth E. Kendall, 2010)

4.3.4. Produk Kerja Pengenalan Lanjutan

Produk kerja dari pengenalan lanjutan adalah (Roger S. Pressman, 2012) :

1. Pernyataan tentang kebutuhan – kebutuhan sistem dan kelayakannya.
2. Pernyataan lengkap tentang lingkup sistem dan produk.
3. Daftar para pelanggan , pengguna , orang – orang lain yang berkepentingan , yang berpartisipasi dalam pengenalan lanjut untuk mendapatkan kebutuhan – kebutuhan sistem.
4. Deskripsi yang berkaitan dengan lingkungan teknis yang dimiliki sistem
5. Daftar kebutuhan – kebutuhan dan batasan – batasan yang dapat diterapkan pada masing – masing kebutuhan.
6. Skenario – skenario pengguna yang menyediakan wawasan – wawasan yang berkaitan dengan penggunaan sistem atau produk saat produk di eksekusi di bawah berbagai kondisi operasi yang berbeda.
7. Semua prototipe yang dikembangkan untuk dapat mendefinisikan kebutuhan – kebutuhan sistem dengan cara yang lebih baik.

4.4. Mengembangkan Use Case

Use case pada dasarnya memperlihatkan sistem atau perangkat lunak dari sudut pandang para pengguna akhir. Use case selalui di definisikan dari sudut pandang aktor – aktor yang

berinteraksi dengan perangkat lunak yang sedang dikembangkan. Sebuah aktor pada dasarnya merupakan peran yang orang – orang saat mereka berinteraksi dengan perangkat lunak. langkah pertama yang harus dilakukan saat hendak menuliskan sebuah use case adalah mendefinisikan sejumlah aktor yang terlibat dalam cerita. Aktor merupakan sejumlah orang yang berbeda yang menggunakan produk dalam konteks fungsi – fungsi dan perilaku – perilaku yang harus dideskripsikan. Perlu di catat , aktor dan pengguna adalah sesuatu yang berbeda. Seorang pengguna mungkin saja memainkan sejumlah peran yang berbeda saat pengguna yang bersangkutan menggunakan perangkat lunak , sementara sebuah aktor merepresentasikan sebuah kelas eksternal yang memainkan suatu peran dalam konteks suatu kasus pengguna. Saat aktor – aktor teridentifikasi dengan baik , use case dapat dikembangkan. Pada kebanyakan kasus use case dielaborasi lebih lanjut sehingga pengembang mendapatkan lebih banyak rincian tentang interaksi – interaksi yang mungkin terjadi antara aktor dengan perangkat lunak yang akan dikembangkan

Deskripsi rincian tentang suatu use case mencakup :

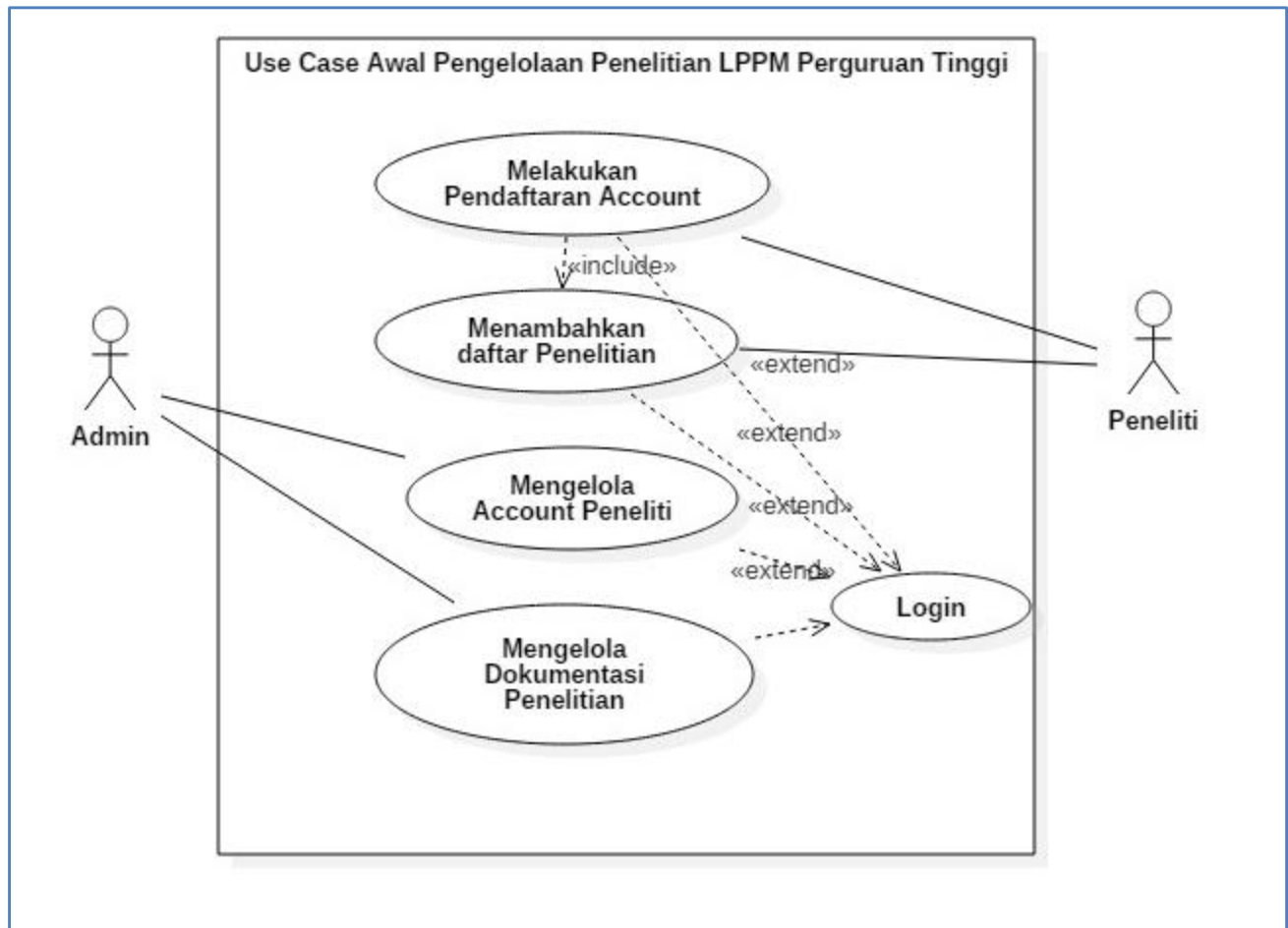
Tabel 4.I Contoh deskripsi use case

| | | |
|----|----------------------|---|
| 1. | Use case name | Pengelolaan Penelitian di LPPM |
| 2. | Aktor Primer | Peneliti |
| 3. | Sasaran dalam kontek | Untuk melakukan penyimpanan dokumentasi penelitian, |

| | |
|---------------|---|
| | sehingga dokumentasi penelitian dapat di publish pada portal <i>research community</i> |
| 4. Prakondisi | Aktor telah diberikan akses berupa account untuk login memasuki portal |
| 5. Pemicu | Peneliti memutuskan untuk menambahkan dokumentasi penelitian |
| 6. Skenario | <ol style="list-style-type: none"> a. Peneliti membuka link dashboard b. Peneliti memasukkan user dan password c. Peneliti memilih tombol ok d. Peneliti memilih menu add research e. Peneliti mengisi data research f. Peneliti mengupload dokumentasi penelitian g. Peneliti memilih tombol save |

| | |
|---|--|
| <p>7. Pengecualian</p> | <p>a. Link dashboard tidak dapat membuka dashboard : peneliti memeriksa koneksi internet , kemudian peneliti harus mengkoneksikan computer ke internet</p> <p>b. Kata sandi tidak benar : peneliti memasukkan ulang password</p> <p>c. Kata sandi tidak dikenali : sistem memberi tanggapan dengan memberikan peringatan</p> |
| <p>8. Permasalahn – permasalahan yang terbuka</p> | <p>a. Adakah cara peneliti menambahkan dokumentasi research dalam bentuk demo video atau bentuk lain?</p> <p>b. Haruskah sistem menampilkan pesan – pesan teks tambahan untuk</p> |

| | |
|--|---|
| | <p>validasi penyimpanan data atau validasi akses masuk sistem?</p> <p>c. Apakah sistem dapat mendeteksi bahwa data reasearch sudah ada , sehingga tidak ada duplikasi data ?</p> <p>d. Apakah sistem dapat memberikan wadah media untuk komunikasi antar peneliti ?</p> <p>e. Apakah sistem dapat memberikan fasilitas pemberian komentar pada dokumentasi research yang telah di publish ?</p> |
|--|---|



Gambar 4.2 Diagram Use Case Awal Pengelolaan Penelitian LPPM Perguruan Tinggi

4.5. Mengembangkan Model Kebutuhan

Sasaran model analisis sesungguhnya adalah untuk memberikan deskripsi dari ranah informasional , fungsional dan perilaku yang dibutuhkan untuk sistem – sistem yang berbasis komputer. Model analisis merupakan gambaran kebutuhan – kebutuhan pada suatu waktu tertentu. Saat model kebutuhan berkembang, beberapa elemen akan menjadi relatif stabil , menyediakan dasar yang kokoh untuk pekerjaan perancangan yang akan mengikuti tahap analisis ini. Namun

demikian, elemen – elemen lainnya dari model analisis mungkin akan terus mengalami perubahan dimana hal ini menunjukkan bahwa orang - orang yang berkepentingan tidak secara peripurna memahami kebutuhan – kebutuhan sistem yang akan dikembangkan.

4.5.1. Elemen – elemen Model Kebutuhan

Ada beberapa elemen cara yang berbeda untuk mencari spesifikasi kebutuhan – kebutuhan untuk sistem berbasis komputer yang akan dikembangkan. Praktisi – praktisi perangkat lunak percaya bahwa merupakan hal yang sangat baik untuk menggunakan berbagai bentuk representasi yang berbeda untuk mendapatkan model kebutuhan . Bentuk – bentuk representasi yang berbeda pada dasarnya akan memaksa pengembang untuk mempertimbangkan kebutuhan – kebutuhan perangkat lunak dari berbagai sudut pandang yang berbeda yaitu suatu pendekatan yang pada dasarnya memiliki peluang tinggi untuk dapat menyingkapkan pengabaian – pengabaian , ketidakkonsistenan dan ambiguitas. Ada sejumlah elemen yang bersifat generik yang bersifat umum untuk kebanyakan model kebutuhan .

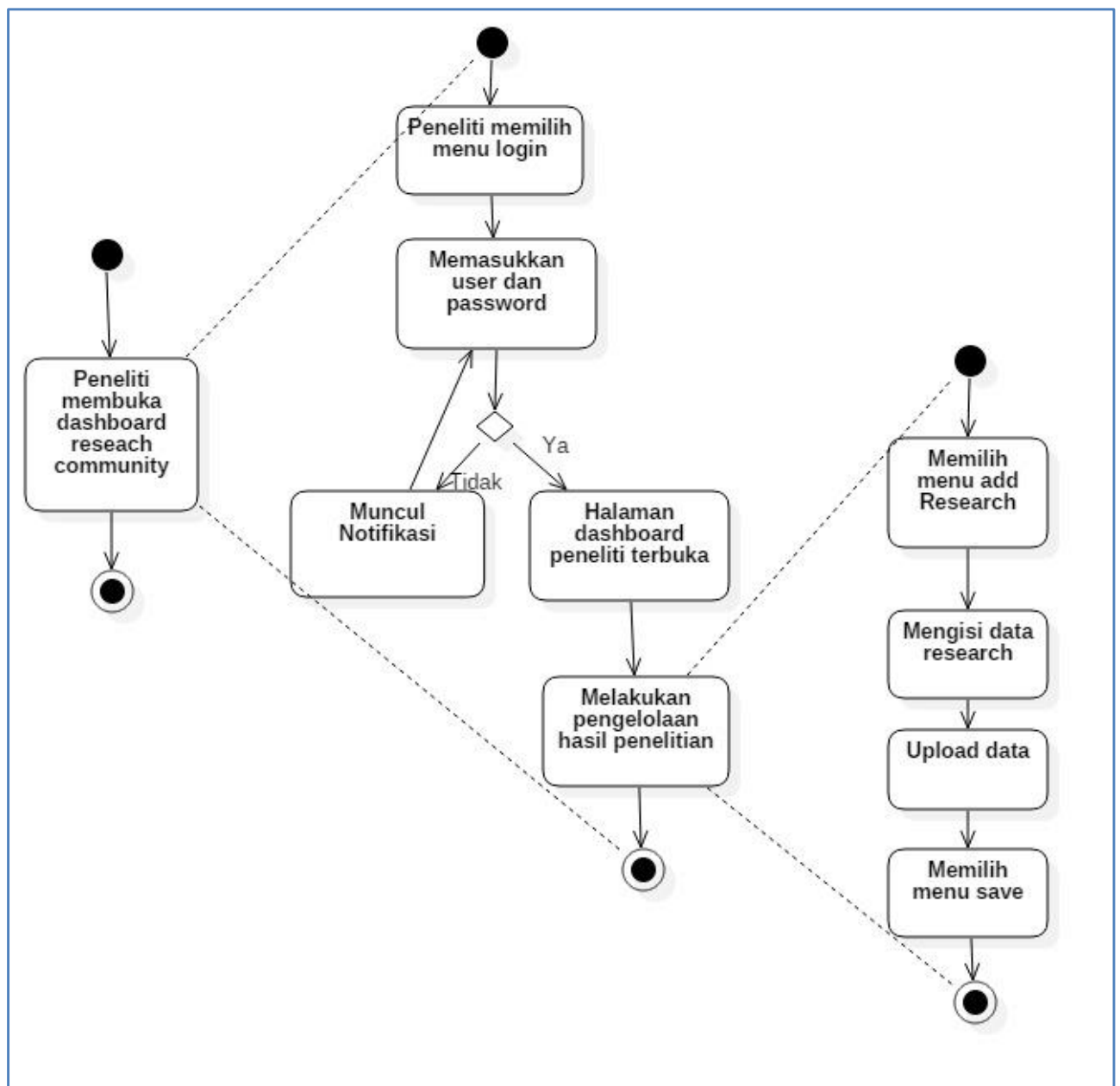
I. Elemen – elemen berbasis skenario

Perangkat lunak dideskripsikan dari sudut pandang pengguna menggunakan pendekatan berbasis skenario. Contohnya adalah use case pada gambar 4.1 yang bersifat dasar dan diagram use case tersebut akan di kembangkan menjadi

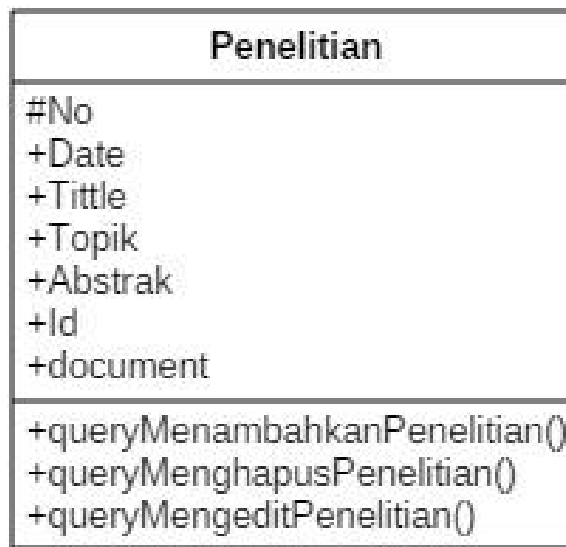
use case – use case berbasis pola yang akan lebih mudah dielaborasi.

2. Elemen – elemen berbasis kelas

Masing – masing skenario penggunaan pada dasarnya mengimplikasikan sejumlah objek yang dapat dimanipulasi oleh perangkat lunak saat sebuah aktor melakukan interaksi dengannya. Objek – objek ini kemudian dapat dikelompokkan ke dalam kelas – kelas yaitu sebuah kumpulan dari sesuatu yang memiliki atribut – atribut yang serupa dan memiliki perilaku – perilaku yang umum sifatnya. Contohnya adalah diagram kelas UML. Diagram kelas itu memperlihatkan daftar atribut dan operasi – operasi yang dapat diterapkan untuk memodifikasi atribut – atribut yang dimiliki kelas tersebut. Sebagai tambahan pada diagram – diagram kelas , elemen – elemen pemodelan dianalisis lainnya memperlihatkan bagaimana kelas – kelas saling berkolaborasi satu dengan lainnya dan juga memperlihatkan relasi – relasi serta interaksi – interaksi yang terjadi di antara suatu kelas dengan kelas – kelas lainnya.



Gambar 4.3 Diagram aktivitas untuk pengelolaan penelitian pada LPPM Perguruan Tinggi



Gambar 4.4 Diagram Class untuk penelitian

3. Elemen – elemen Perilaku

Perilaku suatu sistem berbasis komputer bisa saja memiliki imbas tertentu pada rancangan tertentu yang dipilih dan mungkin juga memiliki imbas tertentu pada pendekatan implementasi yang akan diterapkan. Model – model kebutuhan harus menyediakan elemen – elemen pemodelan yang memperlihatkan perilaku perangkat lunak yang akan dikembangkan. Diagram state UML adalah suatu metode yang memperlihatkan perilaku sistem dengan memperhatikan state-nya . Sebuah state pada dasarnya merupakan setiap modus perilaku yang dapat diobservasi secara eksternal. Diagram state memperlihatkan tindakan – tindakan tertentu yang diambil sebagai akibat terjadinya suatu

event tertentu. Diagram state merepresentasikan perilaku sistem secara keseluruhan.

4. Elemen – elemen berorientasi aliran
Informasi – informasi akan ditransformasi dalam bentuk aliran di dalam sistem berbasis komputer. Sistem menerima masukan – masukan dalam berbagai bentuk, menerapkan fungsi – fungsi untuk melakukan transformasi terhadap masukan – masukan itu dan menghasilkan keluaran – keluaran dalam bentuk beragam.

4.5.2. Pola – pola Analisis

Setiap orang yang telah menyelesaikan rekayasa kebutuhan untuk beberapa proyek perangkat lunak seringkali memperhatikan beberapa masalah yang terjadi berulang kali pada semua proyeknya yang memiliki ranah aplikasi yang bersifat spesifik. Pola – pola analisis ini menghadirkan solusi – solusi dalam ranah aplikasi yang dapat digunakan ulang saat melakukan pemodelan untuk banyak aplikasi.

4.6. Cara Menegosiasikan Kebutuhan – kebutuhan

Dalam konteks rekayasa kebutuhan yang ideal, pengenalan permasalahan, pengenalan permasalahan lebih lanjut dan elaborasi, sering sekali dapat digunakan untuk menentukan spesifikasi kebutuhan pelanggan secara cukup rinci sehingga tim perangkat lunak langsung dapat bekerja ke tahapan – tahapan selanjutnya dalam aktivitas – aktivitas rekayasa perangkat lunak. kenyataannya kita mungkin harus masuk ke

tahap negoisasi dengan satu atau lebih penyandang dana dan orang – orang yang berkepentingan. Sasaran dari negoisasi pada dasarnya adalah untuk mengembangkan suatu perencanaan proyek yang sesuai dengan keinginan para penyandang dana dan orang – orang yang berkepentingan semnetara pada saat yang bersamaan beusaha untuk melakukan penyesuaian – penyesuaian dengan batasan – batasan yang ada yang ditentukan pada tim perangkat lunak.

Boehm dalam (Roger S. Pressman, 2012) mendefinisikan sejumlah aktivitas negoisasi pada saat awal masing – masing iterasi peroses perangkat lunak. Alih – alih merupakan aktivitas komunikasi pelanggan tunggal. Berikut aktivitas yang dapat dilakukan pada negosisasi kebutuhan :

1. Melakukan identifikasi para stakeholder kunci untuk sistem atau subsistem yang akan dikembangkan
2. Menentukan kondisi menang yang akan didapatkan oleh mereka yang berkepentingan
3. Menegoisasikan kondisi menang oarang – orang yang berkepentingan untuk berusaha merekonsiliasikannya ke dalam sejumlah kondisi menang – menang untuk semua orang yang terlibat dalam negoisasi.

4.7. Cara Menvalidasi Kebutuhan – kebutuhan

Saat masing – masing elemen dari model kebutuhan dibuat, elemen – elemen itu harus diperiksa ketidakonsitenannya, harus diperiksa untuk mengetahui adanya pengabaian – pengabain serta ambiguitas. Kebutuhan – kebutuhan tersebut dilakukan tinjauan oleh orang – orang yang berkepentingan.

Suatu tinjauan model kebutuhan seharusnya dapat menjawab pertanyaan – pertanyaan berikut (Roger S. Pressman, 2012) :

1. Apakah masing – masing kebutuhan konsiten dengan sasaran untuk produk secara keseluruhan ?
2. Apakah semua akebutuhan telah dispesifikasi pada peringkat abstraksi yang semestinya ? yaitu apakah kebutuhan – kebutuhan menyediakan peringkat rincian teknis yang tidak sesuai untuk diakomodasi pada tahap ini ?
3. Apakah kebutuhan memang diperlukan atau kebutuhan itu sesungguhnya memperlihatkan fitur – fitur tambahan yang mungkin tidak terlalu penting bagi sasaran sistem ?
4. Apakah masing – masing kebutuhan menyatu dan tidak ambigu ?
5. Apakah masing – masing kebutuhan memiliki pengusul ? yaitu apakah masing – masing sumber kebutuhan itu tercatat ?
6. Apakah ada kebutuhan – kebutuhan tertentu yang bertentangan dengan kebutuhan – kebutuhan lainnya ?
7. Apakah masing – masing kebutuhan dapat dicapai dalam konteks lingkungan teknik yang akan digunakan oleh sistem atau produk yang akan dikembangkan?
8. Apakah masing – masing kebutuhan dapat diuji setelah masing – masing kebutuhan itu diimplementasikan /
9. Apakah model – model kebutuhan merefleksikan informasi, fungsi dan perilaku sistem yang akan dikembangkan dengan cara yang sesuai ?

10. Apakah model – model kebutuhan telah dipartisi sedemikian rupa sehingga kelak dapat secara progresif memberi informasi yang rinci tentang sistem atau perangkat lunak yang akan dikembangkan ?
11. Apakah pola – pola kebutuhan telah digunakan untuk menyederhanakan model – model kebutuhan ? apakah semua pola telah divalidasi secara semestinya ? apakah semua pola konsisten dengan kebutuhan – kebutuhan pelanggan ?

4.8. Metode Pengumpulan Data

4.8.1. Interview

Interview dalam rangka pengumpulan informasi adalah percakapan terarah dengan tujuan khusus menggunakan format tanya jawab. Interview juga merupakan usaha dalam mengeksplorasi kunci masalah interaksi antara manusia dengan komputer , kegunaan sistem , bagaimana menyenangkan sistem dan bagaimana sistem berguna dalam mendukung tugas individu. Berikut adalah langkah – langkah dalam mempersiapkan interview :

1. Pahami latar belakang organisasi , pahami latar belakang orang yang akan diwawancarai dan organisasinya , dengan tujuan analisis dapat menyusun kosa kata yang sesuai dan memaksimalkan waktu dalam wawancara.
2. Tetapkan tujuan wawancara
3. Putuskan siapa yang akan diwawancarai
4. Siapkan orang yang diwawancarai
5. Tentukan jenis dan struktur pertanyaan

4.8.2. Joint Application Design

Merupakan teknik yang memungkinkan pengembang, manajemen dan kelompok pelanggan bekerja sama untuk membangun suatu produk. Pendekatan ini dapat memberikan manfaat yaitu menyingkat waktu , meningkatkan mutu hasil informasi dan menciptakan identifikasi lebih banyak pengguna dengan sistem informasi sebagai hasil proses – proses yang partisipatif.

Berikut adalah kondisi yang memungkinkan penggunaan Joint Application Design :

1. Kelompok pengguna menginginkan sesuatu yang baru untuk penyelesaian masalah yang ada.
2. Budaya organisasi yang mendukung perilaku penyelesaian masalah secara bersama – sama antar pegawai dari level yang berbeda.
3. Penganalisis memprediksi bahwa jumlah ide – ide yang dapat dihasilkan melalui wawancara empat – mata tidak sebanyak ide yang dihasilkan dari perluasan pengamatan kelompok.

Kelebihan dari penggunaan Joint Application Design adalah :

1. Menghemat waktu wawancara tradisional empat mata
2. Memungkinkan perkembangan yang cepat.
3. Pengembangan desain yang lebih kreatif.

Kekurangan dari penggunaan Joint Application Design adalah :

1. Joint Application Dsign membutuhkan komitmen waktu
2. Jika persiapan setiap sesi Joint Application Design tidak cukup memadai , atau bila laporan tindak lanjut serta dokumentasi untuk spesifikasi – spesifikasi tertentu tidak lengkap, membuat desain menjadi kurang memuaskan.
3. Keahlian – keahlian organisasi dan budaya yang diperlukan tidak cukup untuk dapat dikembangkan sehingga memungkinkan upaya – upaya bersama yang lebih produktif dalam menyusun Joint Application Design.

4.8.3. Menggunakan Kuesioner

Penggunaan kuesioner merupakan teknik pengumpulan informasi yang memungkinkan analis sistem untuk mempelajari sikap , keyakinan , perilaku dan karakteristik dari beberapa orang kunci dalam organisasi yang mungkin dipengaruhi oleh sistem saat ini dan yang diusulkan. Sikap adalah apa yang orang – orang dalam organisasi inginkan dalam hal ini sistem baru. Keyakinan adalah apa yang orang anggap benar sedangkan perilaku adalah apa yang anggota organisasi lakukan. Karakteristik adalah sifat orang atau benda.

Berikut adalah contoh kuesioner (Kenneth E. Kendall, 2010):

53. What are the most frequent problems you experience with computer output?

A. _____

B. _____

C. _____

54. Of the problems you listed above, what is the single most troublesome?

55. Why?

Below are questions about yourself. Please fill in the blanks to the best of your ability.

67. How long have you worked for this company?
_____ Years and _____ Months

68. How long have you worked in the same industry?
_____ Years and _____ Months

69. In what other industries have you worked?

Open-ended questions can ask the respondent for lists ...

... or detailed responses ...

... or short answers.

Gambar 4.4 Contoh kuesioner