

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Definisi Pengadaan Barang dan Jasa**

Pengadaan barang dan jasa merupakan suatu kegiatan pengadaan dalam hal untuk mendapatkan barang dan jasa. Tahap-tahap dalam pengadaan barang dan jasa dengan prakualifikasi yaitu : (Keppres No 54, 2010)

1. pengumuman prakualifikasi
2. pengambilan dokumen prakualifikasi
3. pemasukan dokumen prakualifikasi
4. evaluasi dokumen prakualifikasi
5. penetapan hasil prakualifikasi x
6. pengumuman hasil prakualifikasi
7. masa sanggah prakualifikasi
8. undangan kepada peserta yang lulus prakualifikasi
9. pengambilan dokumen lelang umum
10. penjelasan
11. penyusunan berita acara penjelasan dokumen lelang dan perubahannya
12. pemasukan penawaran
13. pembukaan penawaran
14. evaluasi penawaran
15. penetapan pemenang
16. pengumuman pemenang
17. masa sanggah

18. penunjukan pemenang
19. penandatanganan kontrak

## **2.2 Definisi E-Auction**

Secara umum, *e-auction* sebagai proses pengadaan barang dan jasa yang dilakukan dengan berbasis internet sehingga proses transaksi antara buyer dan supplier dilakukan secara online dengan cepat, aman, dan mudah. *E-Auction*. Ini adalah aplikasi untuk membantu proses lelang. Pada proses pembelian, lelang dilakukan oleh pembeli dengan mengumpulkan calon-calon supplier. Mereka sebelumnya sudah diberi tahu oleh pembeli tentang jumlah, spesifikasi, dan waktu kebutuhan suatu barang atau jasa. Mereka akan mengajukan penawaran (secara elektronik) dan selama proses lelang mereka biasa merevisi (menurunkan) harga penawarannya. Supplier yang memberikan penawaran terendah pada akhir periode lelang akan keluar sebagai pemenang. (Hermawan, 2008:9)

## **2.3 Landasan Hukum Pengadaan Barang di Indonesia**

Yang menjadi landasan hukum dalam pelaksanaan pengadaan barang / jasa secara elektronik di Indonesia

### **2.3.1 Dasar Hukum**

1. Keppres No. 54 Tahun 2010, tentang Pedoman Pelaksanaan Pengadaan Barang dan Jasa Pemerintah
2. Inpres No. 5 Tahun 2003, tentang Paket Kebijakan Ekonomi Menjelang dan Sesudah Berakhimya Program Kerjasama Dengan International Monetary Fund
3. Inpres No. 5 Tahun 2004, tentang Percepatan Pemberantasan Korupsi

4. Perpres No. 8 tahun 2006, tentang Perubahan Keempat atas Keputusan Presiden No. 8 Tahun 2003 (tentang Pedoman Pelaksanaan Pengadaan Barang/Jasa Pemerintah)

### 2.3.2 Peraturan Perundangan

1. Instruksi Presiden Nomor 3 Tahun 2003 Tentang Kebijakan dan Strategi Pengembangan *e-Government*

Lamp. 1 : Menteri dalam melakukan pematapan *e-Gov* perlu membuat situs transaksi elektronik dalam pelayanan publik.

2. Keputusan Presiden Nomor SO Tahun 2003 Tentang Pedoman Pelaksanaan Pengadaan Barang/Jasa Instansi Pemerintah.

pasal 3 : Pengadaan barang/jasa wajib menerapkan prinsip-prinsip - terbuka dan bersaing.

pasal 10 : Panitia/Pejabat pengadaan harus mengumumkan pengadaan barang/jasa melalui media cetak dan papan pengumuman resmi untuk penerangan umum dan jika memungkinkan melalui media elektronik.

Surat Edaran Menteri PU No. 21/SE/M/2007, Perihal Penerapan Pengadaan Barang/Jasa Secara Elektronik (*e-auction*) di Lingk, Dep. PU TA 2008 dengan meningkatkan kualitas Pelaksanaan dan Kuantitas Penyebaran Pengertian Inventarisasi.

## 2.4 UML

*Unified Modeling Language* (UML) merupakan bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. Standar spesifikasi UML dijadikan standar defacto oleh OMG (Object Management

Group) pada tahun 1997. UML yang berorientasikan obyek mempunyai beberapa notasi standar. Spesifikasi ini menjadi populer dan standar karena sebelum adanya UML, telah ada berbagai macam spesifikasi yang berbeda. Hal ini menyulitkan komunikasi antar pengembang perangkat lunak. Untuk itu beberapa pengembang spesifikasi yang sangat berpengaruh berkumpul untuk membuat standar baru. UML dirintis oleh Grady Booch, James Rumbaugh pada tahun 1994 dan kemudian Ivar Jacobson.

Menurut perintisnya, UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem.

Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan. Memahami UML itu sebagai bahasa visual itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin. Bahasa visual lebih dekat ke mental model pikiran kita, sehingga pemodelan menggunakan bahasa visual bisa lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

Seperti yang telah dipaparkan di atas, UML yang merupakan turunan dan beberapa metode mempunyai kumpulan diagram grafts sebagai kombinasi dari konsep pemodelan data (entity relationship diagram), pemodelan bisnis (work flow), pemodelan obyek, dan pemodelan komponen. Diagram grafts tersebut merupakan tampilan dari beberapa level abstraksi yang dapat digunakan secara bersama oleh semua proses pada seluruh lifecycle pengembangan software serta

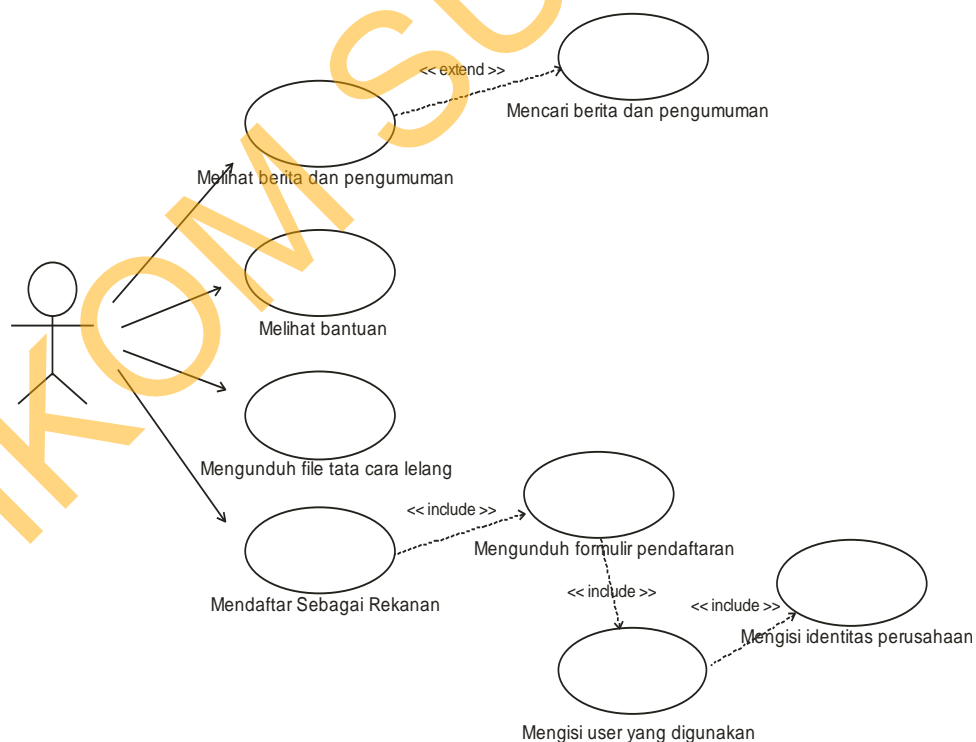
pada implementasi beberapa teknologi yang berbeda. Berikut ini kumpulan diagram graphs yang dipunyai oleh UML. (Sholih, 2005:06)

## 2.5 Use Case Diagram

*Use case diagram* berisi mengenai interaksi antara sekelompok proses dengan sekelompok actor, menggambarkan fungsionalitas dari sebuah sistem yang dibangun dan bagaimana sistem berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan selama proses analisis untuk menangkap kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

*Use Case diagram* terdiri dari 6 elemen pemodelan utama, yaitu *Sistem*, *Actor*, *Use Case*, *Association*, *Dependency*, dan *Generalization*. (Sholih, 2005:07)

Berikut ini merupakan contoh dari *Use Case Diagram* :



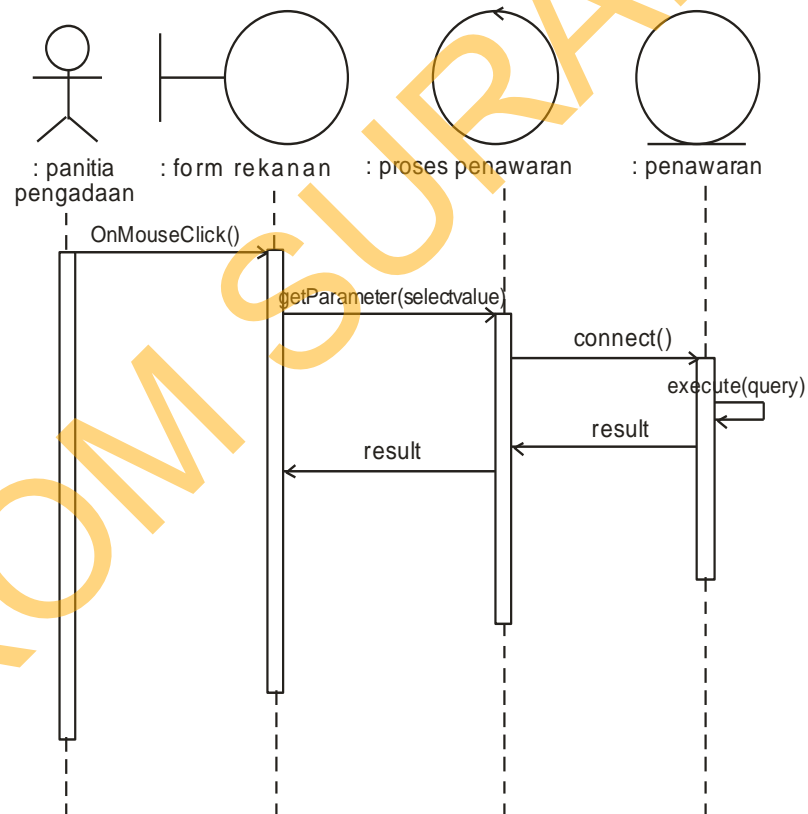
Gambar 2.1. Contoh *Use Case Diagram*

## 2.6 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar (termasuk pengguna, display, dan sebagainya) berupa *message* yang disusun dalam suatu urutan waktu.

Secara khusus, diagram ini berasosiasi dengan *use case*. *Sequence diagram* menggambarkan *behavior* internal sebuah sistem. Dan lebih menekankan pada penyampaian message dengan parameter waktu. (Sholih, 2005:09)

Berikut ini merupakan contoh dari Sequence Diagram :



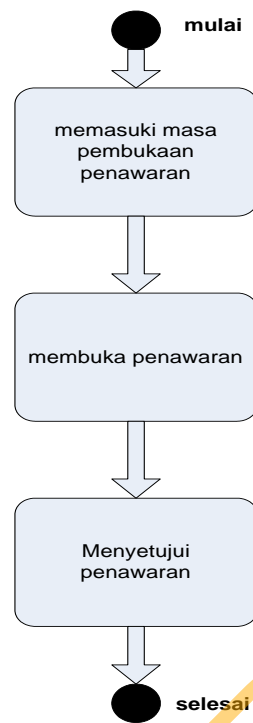
Gambar 2.2. Contoh *Sequence Diagram*

## 2.7 Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state* diagram khusus, di mana sebagian besar *stage* adalah *Action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior* internal sebuah sistem.

Sebuah aktifitas dapat direalisasikan oleh suatu *use case* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana actor menggunakan sistem untuk melakukan aktifitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktifitas. *Decision* digunakan untuk menggambarkan behavior pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertical. *Activity diagram* dapat dibagi menjadi beberapa object *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktifitas tertentu. (Sholih, 2005:08)

Berikut ini merupakan contoh dari *activity diagram* :



Gambar 2.3. Contoh Activity Diagram

## 2.8 Aplikasi Berbasis Web

Sejalan dengan perkembangan dan tuntutan dalam penggunaan web, web server tidak hanya mengirimkan file-file HTML yang isinya statis belaka, namu web server masa kini telah dimanfaatkan untuk mengirim informasi yang sifatnya dinamis, berubah-ubah sesuai kondisi dan untuk memenuhi permintaan pengguna. Sekarang orang biasa membangun aplikasi web untuk meng-*edit* data, memasukkan data, membaca informasi yang disediakan oleh server *database*, maupun sekedar mengirimkan email.

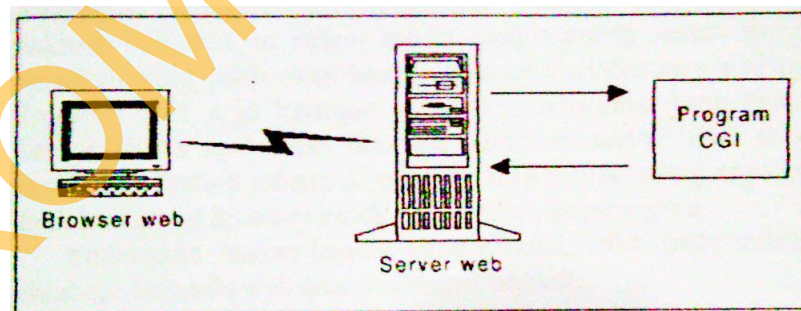
Keuntungan dari aplikasi yang dikembangkan dengan berbasis web adalah pengguna dapat memiliki platform ataupun sistem operasi yang berbeda-beda serta dapat mengakses aplikasi dari mana saja asalkan terhubung dengan jaringan.



Salah satu kekuatan utama yang memungkinkan untuk membangun aplikasi-aplikasi web yang dinamis adalah kemampuan *web server* untuk mengakses sistem database. Web server saat ini bias melakukan koneksi dengan RDBMS (Relational Database Management Sistem) seperti Oracle, SQL Server maupun My SQL dengan mudah.

Untuk keperluan pengembangan aplikasi web yang dinamis ini, pertama kali diperkenalkan CGI (Common Gateway Interface). CGI adalah bagian dari web server yang dapat berkomunikasi dengan program lain diluar web server. CGI memungkinkan web server memanggil suatu program, lalu mengirimkan data-data spesifik dari pengguna program tersebut. Hasil proses diterima oleh CGI yang selanjutnya menyerahkan kepada web server untuk kemudian, mengirimkan informasi tersebut kembali dalam bentuk HTML ke browser web pengguna. (Hermawan, 2008:23)

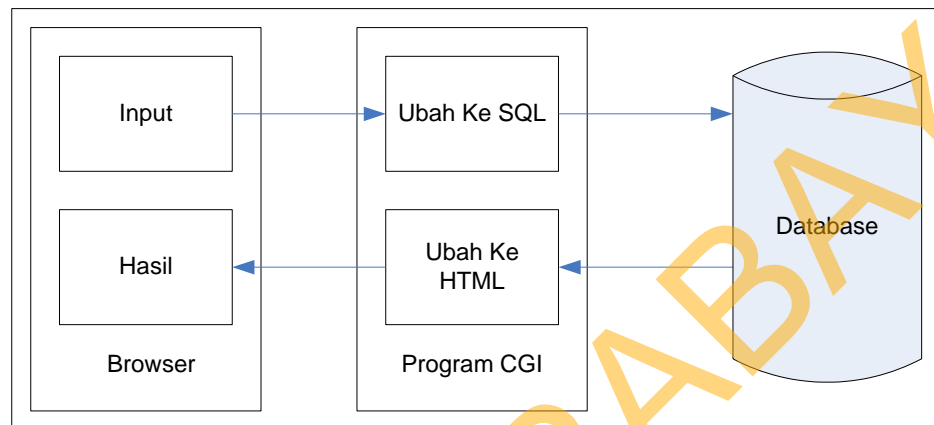
Proses ini dapat digambarkan seperti gambar berikut:



Gambar 2.4. Aplikasi berbasis web

Kebanyakan RDBMS seperti Oracle, SQL server, dan MySQL mendukung bahasa SQL (Structured Query Language) untuk mengakses dan memanipulasi data yang berada dalam sistem database relasional. SQL memungkinkan pengaksesan data secara mudah, cepat dan tidak tergantung pada

platform sistemnya. Program CGI memanfaatkan SQL ini untuk mengakses dan memanipulasi data dalam sistem database. Gambar berikut menggambarkan prinsip kerja pengaksesan *database* melalui CGI. (Hermawan, 2008:24)



Gambar 2.5. Pengaksesan database melalui CGI

Dari gambar diatas, terlihat bahwa fungsi CGI pada *web server* adalah menghasilkan *query* (perintah untuk mengakses database) dalam bentuk SQL kepada sistem RDBMS untuk membaca, memasukkan, menghapus, ataupun mengubah data tertentu dari dan ke dalam media penyimpanan sesuai dengan masukan yang telah ditentukan oleh *user* dari *browser web-nya*. Program CGI juga bertugas mengubah informasi menjadi HTML dan selanjutnya bisa dikirim lagi ke *browser* untuk ditampilkan ke pengguna.

Sedangkan bagian-bagian yang dimiliki oleh suatu aplikasi database berbasis web adalah sebagai berikut:

- a. Client : web browser dari user, java applet, aplikasi java, flash, atau platform dari program *client* yang berdiri sendiri.
- b. Aplikasi Logika : algoritma pengkodean pada *script* CGI, modul khusus dari *web server*, atau aplikasi server yang berdiri sendiri.

- c. Konektifitas *database* : API dari database, protocol konektifitas umum seperti ODBC atau JDBC.
- d. Server untuk *database* : RDBMS, ODBMS, dan lain-lain.

Implementasi dari beberapa aplikasi dapat menggunakan model *multi-user*, karena satu atau beberapa lapisan dapat dipakai bersama-sama. Namun implementasi secara umum yang biasa dipakai adalah sistem *three-tier* (tiga lapisan), yang terdiri atas tiga komponen utama yaitu:

1. Lapisan pertama, yang merupakan aplikasi dari client, contohnya: browser dari user.
2. Lapisan kedua, merupakan aplikasi Web server, Script CGI dan API koneksi untuk database, seperti Apache Web Server dengan modul PHPnya, yang mendukung database MySQL, dan data sript PHP.
3. Lapisan ketiga adalah server untuk database.