

II. TINJAUAN PUSTAKA

A. Sistem Informasi

Sistem merupakan serangkaian aturan atau prosedur yang telah ditetapkan dan dijalankan dalam rangka mencapai suatu tujuan. Sistem adalah hubungan antara unit satu dengan unit lainnya yang saling berhubungan satu sama lainnya dan yang tidak dapat dipisahkan menjadi suatu kesatuan dalam mencapai tujuan yang ditetapkan (Gaol, 2008).

Informasi adalah kumpulan fakta terorganisir dan sudah diproses sehingga memiliki nilai tambahan di luar nilai fakta itu sendiri. Data terdiri dari fakta-fakta mentah, seperti nomor karyawan, jumlah jam kerja atau pesanan penjualan (Stair dan Reynolds, 2011). Data merupakan nilai, keadaan atau sifat yang berdiri sendiri dan lepas dari konteks apapun (Fatta, 2007).

Sistem informasi adalah seperangkat unsur yang saling terkait atau komponen yang mengumpulkan (*input*), memanipulasi (*proses*), menyimpan, dan menyebarkan (*output*) data dan informasi, memberikan reaksi korektif (umpan balik) untuk memenuhi tujuan (Stair dan Reynolds, 2011). Jogiyanto dalam Nugroho (2012), sistem informasi adalah suatu sistem yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan

laporan-laporan yang diperlukan oleh pihak luar yang berkaitan. Andriyani (2009) mengatakan sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi.

Muflich (2011) dalam penelitiannya mengatakan ada 6 komponen dari sebuah sistem informasi yaitu, perangkat keras (*hardware*), perangkat lunak (*software*), basis data (*database*), jaringan (*network*), prosedur dan orang. Seluruh komponen tersebut menjadi sebuah kesatuan dalam sebuah sistem informasi.

B. Dosen

Peraturan Pemerintah Republik Indonesia Nomor 41 Tahun 2009 tentang Tunjangan Profesi Guru dan Dosen, Tunjangan Khusus Guru dan Dosen, Serta Tunjangan Kehormatan Profesor, Bab I Ketentuan Umum Pasal 1 ayat (2) Dosen adalah pendidik profesional dan ilmuwan dengan tugas utama mentransformasikan, mengembangkan, dan menyebarkan ilmu pengetahuan, teknologi, dan seni melalui pendidikan, penelitian, dan pengabdian kepada masyarakat. Sembiring (2008), dosen adalah salah satu komponen manusiawi dalam proses belajar, yang ikut berperan dalam usaha pembentukan sumber daya manusia yang potensial di bidang pembangunan.

Tiga tugas utama dosen dalam tridharma perguruan tinggi yaitu menyelenggarakan pendidikan, melakukan penelitian dan melakukan

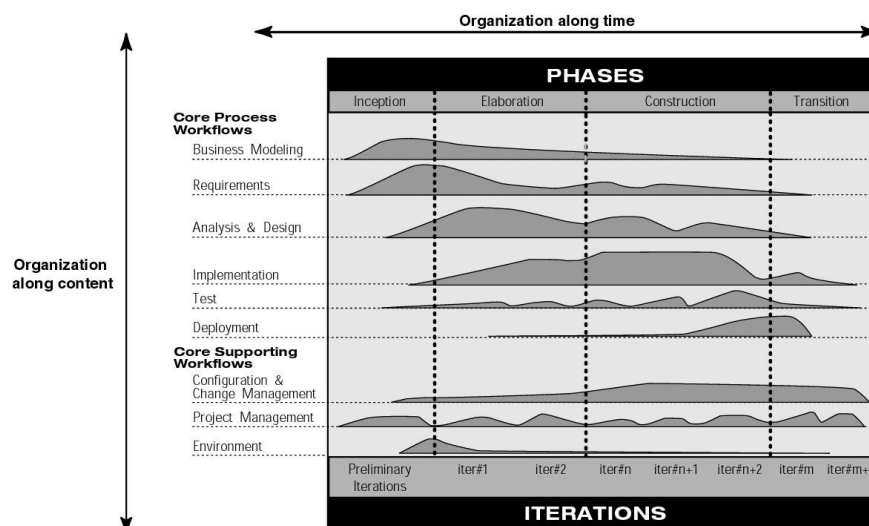
pengabdian kepada masyarakat. Tugas dosen dalam menyelenggarakan pendidikan dapat berupa melaksanakan perkuliahan atau tutorial, menyelenggarakan kegiatan pendidikan di laboratorium, membimbing tugas akhir mahasiswa, menjadi pembimbing Kuliah Kerja Nyata (KKN) dan atau Kerja Praktik (KP), membina kegiatan kemahasiswaan di bidang akademik, mengembangkan program perkuliahan, menyampaikan orasi ilmiah dan mengembangkan bahan ajar berupa buku atau artikel ilmiah. Tugas dosen dalam melakukan penelitian merupakan tugas di bidang penelitian dan pengembangan karya ilmiah berupa menghasilkan karya penelitian, menerjemahkan atau menyadur buku ilmiah, menyunting karya ilmiah, membuat rancangan dan karya teknologi dan membuat rancangan karya seni. Tugas dalam pengabdian kepada masyarakat dapat berupa melakukan pengembangan hasil pendidikan dan penelitian yang dapat dimanfaatkan oleh masyarakat, memberi pelatihan, penyuluhan dan penataran kepada masyarakat dan membuat karya pengabdian kepada masyarakat. Tugas lain yang menunjang tiga tugas utama dosen adalah menjadi anggota dalam suatu panitia atau badan perguruan tinggi atau lembaga pemerintahan, menjadi anggota organisasi profesi, menjadi delegasi nasional pada pertemuan internasional, berperan aktif dalam pertemuan ilmiah, mendapat tanda jasa atau penghargaan, dan menulis buku pelajaran SMA, SMP, dan SD (Tim Penyusun, 2010).

C. Metode *Rational Unified Process* (RUP)

Rational Unified Process (RUP) adalah proses rekayasa perangkat lunak yang menggunakan pendekatan disiplin untuk menetapkan tugas dan tanggung

jawab dalam pengembangan organisasi (Manalil, 2010). RUP adalah sebuah proses pembangunan sistem meliputi seluruh *lifecycle* pembangunan perangkat lunak yang menyediakan suatu pendekatan untuk membantu tugas dan tanggung jawab suatu pembangunan organisasi. RUP diciptakan, dikembangkan dan dikelola oleh *Rational Software* sekarang IBM. Tujuannya adalah menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna dan dapat diprediksi penjadwalan dan biaya pengembangannya (Kruchten, 2003).

Kruchten dalam Fajriyah (2011), RUP baik digunakan karena mendukung 6 (enam) praktik terbaik dalam pengembangan perangkat lunak, yaitu pengembangan perangkat lunak secara iteratif, pengelolaan persyaratan, menggunakan arsitektur berbasis komponen, memvisualisasi pemodelan perangkat lunak, secara terus-menerus memverifikasi kualitas perangkat lunak, dan mengawasi perubahan perangkat lunak.



Gambar 1. Arsitektur *rational unified process* (Sumber: *Rational Unified Process Best Practices for Software Development Teams*)

Aktivitas dalam pengembangan perangkat lunak menggunakan RUP terfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML), karena menggunakan konsep berorientasi objek atau *object oriented*. RUP terdiri dari dua dimensi, seperti pada Gambar 1, dua dimensi dalam RUP yaitu:

- a. Struktur dinamis, menggambarkan aspek dinamis dalam sebuah pengembangan perangkat lunak, yang dijabarkan dalam beberapa fase yang memiliki *major milestone* atau tonggak utama sebagai tanda berakhirnya fase tersebut. Setiap fase dapat terdiri dari satu atau lebih pengulangan atau iterasi. Dimensi horizontal ini terdiri atas *Inception*, *Elaboration*, *Construction* dan *Transition*.
- b. Struktur statis, merupakan dimensi vertikal yang mewakili aspek-aspek statis dari sebuah pengembangan perangkat lunak terdiri dari empat elemen penting, yakni siapa yang melakukan (*who is doing*), apa (*what*), bagaimana (*how*) dan kapan (*when*). Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environtment* (Manalil, 2010).

1. Struktur Dinamis RUP

Struktur dinamis berhubungan dengan siklus pengembangan perangkat lunak dibagi ke dalam empat fase yang berurutan yaitu:

- a. *Inception*, fase awal atau *inception* pengembang mendefinisikan batasan kegiatan, menentukan bisnis proses, melakukan analisis

kebutuhan pengguna, melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*). Fase ini akan menghasilkan proses bisnis, *use case* awal dan rencana pengembangan proyek.

- b. *Elaboration*, tujuan dari fase elaborasi (*elaboration*) adalah untuk menganalisa permasalahan utama, membangun arsitektur dasar, mengembangkan rencana proyek, dan menghilangkan resiko tertinggi dari proyek. Hasil dari fase elaborasi ini adalah sebuah *use case* yang sudah 80%, deskripsi arsitektur perangkat lunak, persyaratan non-fungsional, dan gambaran umum pengembangan perangkat lunak.
- c. *Construction*, fase konstruksi terfokus pada pengembangan perangkat lunak baik komponen utama maupun fitur-fitur pendukung dengan melakukan sederet iterasi. Setiap iterasi terdapat proses analisa, desain, implementasi, dan pengujian. Proses pengembangannya dapat menggunakan konstruksi paralel agar mempercepat hasil perangkat lunak. Hasil yang diharapkan dari fase ini adalah sebuah produk perangkat lunak yang siap digunakan oleh *end-user*, yaitu sebuah produk perangkat lunak yang terintegrasi dengan *platform* yang memadai dan penjelasan dari produk tersebut
- d. *Transition*, fase transisi merupakan fase dimana perangkat lunak hasil dari pengembangan didistribusikan kepada pengguna akhir (*end-user*). Kegiatan-kegiatan yang dilakukan pada fase ini antara lain: *beta testing* untuk memvalidasi sistem baru terhadap harapan pengguna, konversi

database operasional, pelatihan bagi pengguna dan pengelola, dan pemasaran produk (*Rational Software*, 1998).

2. Iterasi

Fase-fase dalam *rational unified process* selanjut dibagi menjadi iterasi. Iterasi adalah *loop* pengembangan yang lengkap sehingga dalam rilisnya (internal atau eksternal) dari *executable* produk, bagian dari produk akhir sedang dikembangkan, menjadi sistem akhir secara bertahap dari iterasi ke iterasi (Manalil, 2010).

3. Alur Kerja

Alur kerja dalam RUP dibagi menjadi dua yaitu alur kerja utama dan alur kerja pendukung. Alur kerja utama dalam RUP antara lain :

- a. *Business modeling*, pemodelan bisnis merupakan tahapan menetapkan proses bisnis dari sistem yang akan dikembangkan. Proses bisnis didokumentasikan ke dalam *business use case*. *Business use case* merupakan analisis untuk memahami bagaimana bisnis harus mendukung proses bisnis yang ada dalam organisasi.
- b. *Requirement*, persyaratan merupakan alat untuk menentukan dan menggambarkan kemampuan sistem. Identifikasi aktor, menyatakan pengguna dan sistem lain yang berinteraksi dengan sistem yang sedang dikembangkan. *Use case* diidentifikasi dan dikembangkan sesuai dengan kebutuhan aktor. *Use case* dideskripsikan secara rinci untuk mengetahui

bagaimana sistem berinteraksi dengan aktor dan berfungsi sebagai benang pemersatu siklus pengembangan sistem.

- c. *Analysis and Design*, analisis dan desain bertujuan untuk membuat rancangan sistem yang akan direalisasikan dalam bentuk implementasi. Hasil dari analisis dan desain adalah model desain dan model analisis opsional. Model desain adalah abstrak dari *source code* program sebagai acuan bagi *programmer* dalam mengimplementasikan sebuah sistem.
- d. *Implementation*, implementasi merupakan proses menerjemahkan desain ke dalam pengembangan sistem. Implementasi bertujuan untuk melakukan pengelompokkan kode, kelas dan objek, menguji komponen yang dikembangkan, dan mengintegrasikan hasil dari pengembangan subsistem menjadi sistem yang utuh.
- e. *Test*, pengujian dilakukan untuk memastikan interaksi antar objek, memastikan kesatuan integrasi dari semua komponen perangkat lunak, memastikan seluruh kebutuhan sudah diimplementasikan, dan mengidentifikasi kekurangan pada perangkat lunak.
- f. *Deployment*, penyebaran merupakan proses memproduksi rilis perangkat lunak sehingga siap digunakan dan didistribusikan.

RUP juga mengenal alur kerja pendukung, adapun alur kerja pendukung dalam RUP antara lain:

- a. *Project management*, manajemen proyek mendefinisikan berbagai macam strategi untuk bekerja dengan proses iterasi.

- b. *Configuration and change management*, mengendalikan perubahan pada pengembangan dan memelihara integrasi hasil pengembangan dan aktifitas manajemen.
- c. *Environment*, mencakup seluruh kebutuhan infrastruktur yang dibutuhkan untuk mengembangkan suatu sistem (Manalil, 2010).

D. Basis Data

Basis data atau *database* adalah kumpulan data yang saling berhubungan. Data merupakan fakta yang dapat direkam dan memiliki makna yang terkandung di dalamnya. Sebuah basis data memiliki sifat implisit antara lain, basis data mewakili beberapa aspek dari dunia nyata yang biasa disebut *miniworld*, basis data merupakan kumpulan hubungan logis dari data dengan beberapa makna yang melekat, dan basis data dirancang, dibangun, dan diisi dengan data untuk tujuan tertentu (Elmasri dan Navathe, 2011). *Database* merupakan kumpulan tabel-tabel yang berisi data yang merupakan kumpulan *field* atau kolom (Anhar, 2010). *Database* adalah aplikasi terpisah yang menyimpan kumpulan data. Setiap *database* memiliki satu atau lebih yang berbeda API untuk membuat, mengakses, mengelola, mencari, dan mereplikasi data yang dimilikinya (Converse *et al.*, 2007).

E. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan. ERD digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya (Wahana

Komputer, 2010). ERD juga digunakan untuk mengidentifikasi data yang akan diambil, disimpan, dan dipanggil kembali (*retrieve*) untuk keperluan tertentu dalam mendukung kegiatan yang dilakukan oleh organisasi. ERD merupakan gambaran struktur dan keterkaitan tabel-tabel data yang menyusun *database* secara detail. ERD merupakan representasi data sebagai entitas, atribut, dan relasi (Marimin *et al.*, 2006).

F. Diagram *Use Case*

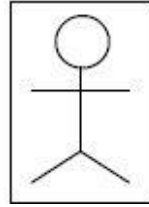
Use case adalah serangkaian langkah-langkah yang saling berhubungan (skenario), baik otomatis maupun manual, dengan tujuan untuk menyelesaikan suatu kegiatan bisnis tunggal. *Use case* menggambarkan fungsi-fungsi sistem dari perspektif pengguna luar. *Use case* adalah hasil dari dekomposisi lingkup fungsi-fungsi dari sistem menjadi statement-statement yang lebih kecil mengenai fungsional oleh fungsi-fungsi sistem. Pembuatan *use case* sudah dibuktikan merupakan suatu teknik yang baik untuk mengerti lebih baik dan mendokumentasi kebutuhan sistem (Dittman, 2000). Diagram *use case* menunjukkan interaksi antara aktor dengan *use case* (Elmasri dan Navathe, 2011).

Suhendar dan Gunadi dalam Sinaga (2013), sebuah *use case diagram* mengandung aktor, *use case*, dan interaksi antara aktor dan *use case*.

1. Aktor

Aktor menggambarkan pengguna *software* aplikasi (*user*). Aktor membantu memberikan suatu gambaran jelas tentang yang harus dikerjakan *software* aplikasi. Sebuah aktor berinteraksi dengan *use case*, tetapi tidak memiliki

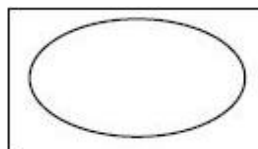
kontrol atas *use case*. Aktor mungkin seorang manusia, sebuah perangkat keras, atau sistem informasi lain. Aktor dinotasikan seperti Gambar 2.



Gambar 2. Aktor

2. *Use case*

Use case menggambarkan perilaku *software* aplikasi, termasuk di dalamnya interaksi antara aktor dengan *software*. *Use case* adalah pola perilaku, urutan transaksi yang berhubungan antara aktor dan *software* aplikasi, dan sistem yang memberikan sesuatu yang bernilai kepada aktor. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan aktor. *Use case* dinotasikan dalam bentuk elips seperti Gambar 3.



Gambar 3. Notasi *use case*

3. *Association*

Association atau asosiasi digunakan untuk menghubungkan aktor dengan *use case*. Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara aktor dan *use case*.



Gambar 4. Notasi asosiasi

Use case dibagi menjadi dua yaitu *use case* konkret dan *use case* abstrak. *Use case* konkret adalah *use case* yang dibuat langsung karena keperluan aktor. *Use case* abstrak adalah *use case* yang tidak pernah berdiri sendiri. *Use case* abstrak senantiasa termasuk di dalam (*include*) dan diperluas dari (*extend*) atau merumpun (*generalize*) *use case* lainnya. Hubungan *include* menggambarkan bahwa suatu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya. Hubungan *extend* antar *use case* berarti bahwa satu *use case* merupakan tambahan fungsionalitas dari *use case* yang lain jika kondisi atau syarat tertentu terpenuhi.

G. Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan dalam penelitian ini antara lain :

1. MySQL
2. XAMPP
3. PHP
4. CSS
5. Notepad++

1. MySQL

MySQL adalah salah satu *database* yang paling mudah pengelolaannya pada semua *platform*, dan karena begitu ringan, dapat dijalankan pada semua komputer bahkan yang berspesifikasi rendah (Converse *et al.*, 2004). MySQL merupakan salah satu sistem *database* yang sangat handal karena menggunakan SQL (*Structured Query Language*). SQL berfungsi sebagai

bahasa penghubung antara program *database* dengan bahasa pemrograman. MySQL merupakan *database* yang sangat cepat, beberapa *user* dapat menggunakan secara bersamaan. MySQL merupakan sistem manajemen *database* dan relasional sistem *database* (Sakur, 2005).

Query adalah pertanyaan yang diajukan untuk mengambil informasi dari *database*. *Query* merupakan bagian dari DML (*Database Manipulation Language*) untuk pengambilan informasi. Jenis-jenis *query* antara lain: *create*, *insert*, *select*, *update*, dan *delete query* (Kristanto, 2004). Contoh *query* yang digunakan dalam MySQL adalah sebagai berikut:

```
create database siksen ;

create table pelaksana (
nama varchar(30),
id_pelaksana char(18) primary key,
email varchar (36),
alamat text,
telepon varchar (14)
);

insert into pelaksana (
'Muhammad Chairuddin', '199109252014121001',
'chairuddin@gmail.com', 'Bandarlampung',
'072100102');

select *from pelaksana where nama='Muhammad Chairuddin';
```

2. XAMPP

XAMPP adalah *installer* yang mengemas *Apache*, PHP, dan MySQL untuk *Windows* dalam satu paket. XAMPP dapat dijadikan *server*, *server* lokal ini disebut *localhost* (Imansyah, 2010). XAMPP adalah sebuah *software* yang berfungsi menjalankan website berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal (Wicaksono, 2008). XAMPP adalah instalasi alternatif bagi administrator yang tidak ingin direpotkan

dengan instalasi *apache* dan PHP yang terpisah, karena pada XAMPP sudah terdapat paket instalasi *Apache* dan PHP yang sudah terintegrasi dan terkonfigurasi otomatis sehingga mudah digunakan (Utomo *et al.*, 2012).

3. PHP

PHP adalah bahasa pemrograman *web server-side* yang bersifat *open source* dan merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*) (Anhar, 2010). Sebuah gagasan dari seorang programmer bahasa C asal Denmark, Rasmus Lerdorf pada tahun 1995 merupakan awal dari lahirnya bahasa pemrograman PHP. PHP (*Hypertext Preprocessor*) merupakan pemrograman berbasis *web* yang memiliki kemampuan untuk memroses dan mengolah data secara dinamis.

Penulisan *script* diawali dengan `<?php` dan diakhiri dengan `?>` dan setiap perintah diakhiri dengan `;` dan *script* PHP dapat dikombinasikan dengan kode HTML. Contoh *script* PHP dan *script* PHP yang dikombinasikan dengan kode HTML adalah:

```
<?php
    echo "Script PHP";
?>
```

Kode Program 1. Contoh *Script* PHP

```
<html>
<title>Sistem Informasi Penelitian dan Pengabdian
Dosen</title>
<body>
<?php
    echo "Script PHP";
?>
</body>
</html>
```

Kode Program 2. Contoh kombinasi *Script* PHP dengan kode HTML

Beberapa *script* PHP yang digunakan pada penelitian ini antara lain:

a. *Delimiter*

```
<?php .....; ?>
```

Kode Program 3. *Delimiter*

Delimiter digunakan untuk memulai dan mengkhiri dokumen PHP.

b. Fungsi *include*

```
<?php
    include 'koneksi.php';
?>
```

Kode Program 4. *Include*

Fungsi *include* digunakan untuk menyertakan dokumen PHP ke dalam suatu program PHP.

c. Variabel

```
<?php
    $batas = 10;
?>
```

Kode Program 5. Variabel

Variabel adalah sebuah istilah untuk menyatakan tempat atau wadah untuk menampung nilai sementara.

4. CSS

Casscanding Style Sheet atau CSS adalah dokumen yang berdiri sendiri dan dapat di masukkan dalam kode HTML atau sekedar menjadi rujukan oleh HTML dalam pendefinisian *style*. CSS menggunakan kode-kode yang tersusun untuk menetapkan *style* pada elemen HTML atau dapat juga

digunakan untuk membuat *style* baru yang biasa disebut *class* (Sulistiyawan *et al.*, 2008).

Ollie (2008), CSS adalah pelengkap HTML, CSS digunakan untuk menentukan warna, *font*, *layout*, dan berbagai macam hal yang menyangkut tampilan. Rumus inti sintaks CSS adalah *selector {fproperty:valueg}*, contohnya seperti berikut.

```
p{font-family : "Times New Roman"}
```

Kode Program 6. Contoh *Script* CSS

Artinya jenis huruf yang digunakan dalam paragraf adalah *Times New Roman*.

5. Notepad++

Notepad++ adalah sebuah program *editor* teks, hampir mirip dengan Notepad yang ada pada *Windows*, juga digunakan untuk *source code editor* yang memiliki kelebihan pada tampilannya antara lain *syntax coloring*, *highlight*, dan *folding* (Haromainy dan Creativity, 2010).

H. *Heuristic Evaluation*

Heuristic Evaluation (HE) merupakan metode pengujian perangkat lunak dengan mengukur tingkat kenyamanan pengguna dalam interaksi manusia dengan komputer. Metode ini pertama kali dikenalkan dan digunakan oleh Nielsen dan Molich pada tahun 1990. HE bertujuan untuk mengidentifikasi masalah yang ada pada desain antar muka *website* (Purnamasari *et al.*, 2012).

Pengujian menggunakan metode HE menggunakan 13 prinsip atau aspek yaitu *Visibility of system status, Match with the real world, User Control and Freedom, Consistency and standards, Help users recognize, diagnose, and recover from errors, Error prevention, Recognition than recall, Flexibility and efficiency of use, Aesthetic and minimalist design, Help and documentation, skills, Pleasurable and Respectful Interaction with the User, dan privacy* (Xerox, 1995).