

BAB II LANDASAN TEORI

2.1 Konsep Dasar Teori

2.1.1. Sistem

Pendekatan sistem yang lebih menekankan pada prosedur oleh Jogiyanto H.M (2001 : 1) didefinisikan sebagai berikut : “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.

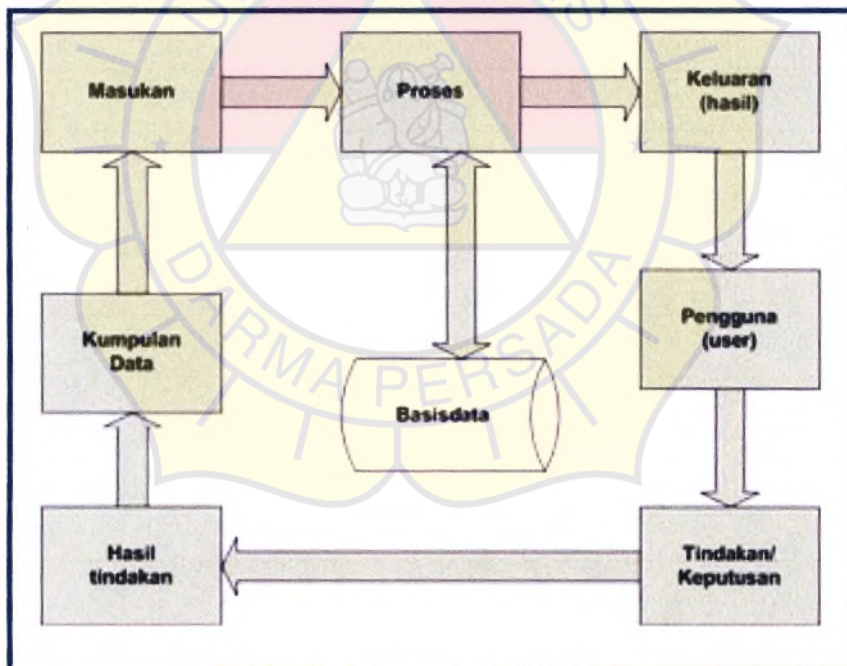
Pendekatan sistem yang lebih menerapkan pada elemen atau komponennya oleh Jogiyanto H.M (2001 : 2) didefinisikan sebagai berikut : “Sistem adalah kumpulan dari elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu. Sistem dapat diklasifikasikan dari beberapa sudut pandang di antaranya:

1. Sistem abstrak adalah suatu sistem yang berupa pikiran atau ide-ide yang tidak nampak secara fisik. Sedangkan sistem fisik adalah suatu sistem yang nampak secara fisik dan dapat dilihat.
2. Sistem alamiah dan sistem yang terjadi melalui proses alam, bukan buatan manusia. Sedangkan sistem buatan manusia adalah suatu sistem yang melibatkan antara manusia dan mesin.
3. Sistem tertentu dan sistem tidak tertentu (deterministik) beroperasi dengan cara yang dapat diramalkan secara tepat, interaksi antara bagian-bagian diketahui dengan pasti. Bila seseorang memiliki uraian keadaan sistem pada saat tertentu beserta uraian operasinya maka keadaan sistem

selanjutnya dapat disebut secara tepat tanpa kesalahan. Sebagai contoh program komputer yang melaksanakan secara tepat sesuai dengan rangkaian dengan instruksi-instruksi yang diberikan. Sistem tidak tertentu (probalistik) dapat diuraikan dalam istilah perilaku yang mungkin tetapi selalu ada sistem kesalahan atas ramalan atas ramalan terhadap jalannya sistem. Sistem persediaan barang adalah sistem probalistik.

2.1.2. Sistem tertutup

Sistem Tertutup adalah sistem yang tidak berhubungan dengan lingkungan lainnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luarnya.



Gambar 2.1 Siklus Sistem Informasi

2.2. Informasi

Informasi ibarat darah yang mengalir dalam tubuh suatu organisasi, sehingga informasi ini sangat penting bagi organisasi. Jogiyanto H.M (2005:8) mendefinisikan sebagai berikut : “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya.

Dalam siklus informasi data merupakan yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut agar dapat lebih berarti dan berguna dalam bentuk informasi. Oleh karena itu perlu diolah dengan melalui suatu model proses tertentu. Data yang diolah menjadi informasi akan dapat melahirkan suatu keputusan untuk melakukan tindakan dan seterusnya membentuk siklus yang menurut Jogiyanto dalam bukunya disebut sebagai siklus informasi (*Information Cycle*)

Kualitas dari suatu informasi (*Quality of Information*) tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timeliness*) dan adanya relevansi (*relevance*).

Akurat, berarti informasi harus bebas dari kesalahan-kesalahan tidak biasa atau mengikat. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah bahkan merusak sistem informasi tersebut.

Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Jadi informasi merupakan landasan didalam pengambilan suatu keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal bagi

organisasi. Dewasa ini mahal nya nilai informasi disebabkan cepat nya informasi tersebut diterima, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevan informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.3 Sistem informasi

Dalam suatu instansi atau perusahaan, sistem informasi merupakan jantung dari semua kegiatan manajemen. Juga telah dijelaskan bahwa informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan suatu keputusan. Sistem informasi diperlukan mulai dari perencanaan, operasi, pemeliharaan hingga pengendalian. Dalam proses perencanaan sistem informasi diperlukan model perencanaan, data masuk, dan simulasi model berupa sistem konversi untuk mengubah masukan menjadi suatu keluaran yang dapat dijadikan penunjang dalam pengambilan suatu keputusan bagi pihak manajemen.

Sistem Informasi dikemukakan oleh Henry C. Lucas dalam Jogiyanto H.M (2001:35) bahwa suatu sistem informasi adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian suatu organisasi.

Menurut John F. Nash dan Martin B. Roberts dalam Jogiyanto H.M (2001:35-36) bahwa sistem informasi merupakan suatu dari orang-orang fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal

kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan yang menyediakan suatu dasar untuk pengambilan keputusan.

Menurut James B. Bower, Robert E. Schlosser dan Maurice S. Newman dalam Jogiyanto H.M (2001:36) bahwa sistem informasi adalah suatu cara yang sudah tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan.

Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis dalam Jogiyanto HM (2011:3) "Konsep dari sistem informasi adalah suatu sistem dalam orang yang memperhatikan kebutuhan pengolahan transaksi harian, yang mendukung operasi bersifat manajerial dan kegiatan strategis dari suatu organisasi dengan pihak-pihak tertentu, dengan menghasilkan laporan-laporan yang diperlukan".

2.4. UML

2.4.1 Definisi UML

Unified Modelling Language (UML) menurut Rizki Fauzi (2010:30) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman

apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

2.4.2 Model Model diagram dalam UML

a. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Rizki Fauzi (2010:30).

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri.

Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

b. Activity Diagram

Menurut Raymond McLeod, Jr., George Schell (2001:9) Activity diagrams menggambarkan berbagai alir aktivitas dalam system 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 state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses proses paralel (fork dan 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 aktivitas tertentu.

c. Sequence Diagram

Menurut Jogiyanto H.M (2001:11) Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.

d. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class

dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain

2.5 Sistem Informasi Manajemen

Sistem Informasi Manajemen (*management information system*) merupakan penerapan sistem informasi di dalam organisasi untuk mendukung informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen. Pengertian SIM (Sistem Informasi Manajemen) Kutipan dari Hartono, Jogiyanto. 1999, Yogyakarta : Andi, George M. Scott (2001:05) mendefinisikan "*SIM adalah kumpulan dari interaksi-interaksi sistem-sistem informasi yang menyediakan informasi baik untuk kebutuhan manajerial maupun kebutuhan operasi.*"

Adapun pengertian lain SIM yang didefinisikan oleh Barry E. Cushing Barry E. Cushing (2005;6) sebagai berikut : "*SIM adalah kumpulan dari manusia dan sumber-sumber daya modal di dalam suatu organisasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menghasilkan informasi yang berguna.*"

Dari beberapa definisi tersebut dapat disimpulkan bahwa SIM merupakan suatu sistem yang melakukan fungsi-fungsi untuk menyediakan semua informasi yang mempengaruhi semua operasi organisasi yang berguna untuk meningkatkan manajemen

2.6 Penjualan

Penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana-rencana strategis yang diarahkan pada usaha pemuasan kebutuhan keinginan pembeli, guna mendapatkan penjualan yang menghasilkan laba. Marwan (1991:1)

Penjualan merupakan sumber hidup suatu perusahaan, karena dari penjualan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan untuk mengetahui daya tarik mereka sehingga dapat mengetahui hasil produk yang dihasilkan. Menurut Winardi (1982:3), penjualan adalah suatu transfer hak atas benda-benda. Dari penjelasan tersebut dalam memindahkan atau mentransfer barang dan jasa diperlukan orang-orang yang bekerja dibidang penjualan seperti pelaksanaan dagang, agen, wakil pelayanan dan wakil pemasaran.

Berikut tujuan dari pembuatan aplikasi penjualan tersebut :

1. Memaksimalkan pelayanan bagi konsumen
2. Meminimumkan investasi pada persediaan
3. Perencanaan kapasitas
4. Penyimpanan dan pergerakan material

2.7 E-Commerce

Bryan A. Garner menyatakan bahwa "*E-Commerce the practice of buying and selling goods and services through online consumer services on the Internet. The e, as horted from electronic, has become a popular prefix for other terms associated with* Definisi bagan alir menurut Jogiyanto H.M (2001:56).

Perspektif proses bisnis, E-Commerce adalah aplikasi dari teknologi yang menuju otomatisasi dari transaksi bisnis dan aliran kerja. Dari perspektif layanan,

E-Commerce merupakan suatu alat yang memenuhi keinginan perusahaan, konsumen, dan manajemen untuk memangkas biaya layanan ketika meningkatkan kualitas barang dan meningkatkan kecepatan pengiriman.

Dari perspektif online, E-Commerce menyediakan kemampuan untuk membeli dan menjual barang ataupun informasi melalui *Internet* dan sarana online lainnya.

Tapi dalam penulisan Tugas Akhir ini penulis mengambil pengertian *Ecommerce* sebagai alat/cara modern untuk lebih memperkenalkan sebuah usaha kepada masyarakat melalui kecanggihan teknologi *Internet*.

2.8 PHP

Didefinisikan oleh Muhammad Sadeli (2011:3) Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat server-side scripting. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. PHP dapat dijalankan pada berbagai macam Operating System (OS), misalnya Windows, Linux dan Mac OS. Selain Apache. Sistem manajemen database yang sering digunakan bersama PHP adalah MySQL. PHP mendukung penuh Object Oriented Programming (OOP), integrasi XML, mendukung semua ekstensi terbaru MySQL, pengembangan web services dengan SOAP dan REST, serta ratusan kemampuan. Sama dengan web server lainnya PHP juga bersifat open source sehingga setiap orang dapat menggunakannya dengan gratis. Muhammad Sadeli (2011:5)

2.9 My SQL

MySQL merupakan *software database open source* yang paling populer didunia, dimana saat ini digunakan lebih dari 100 juta pengguna di seluruh dunia. Dengan kehandalan, kecepatan dan kemudahan penggunaannya, MySQL menjadipilihan utama bagi banyak pengembang *software* dan aplikasi baik di *platformweb* maupun *desktop*. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti Yahoo!, Alcatel-Lucent, Google, Nokia, Youtube, Wordpress dan Facebook jugamerupakan pengguna MySQL. (Achmad Solihin 2010;1)

MySQL pertama kali dibuat dan dikembangkan di Swedia, yaitu oleh David Axmark, Allan Larsson dan Michael "Monty" Widenius. Mereka mengembangkan MySQL sejak tahun 1980-an. Saat ini versi MySQL yang sudah stabil mencapaiversi 5x, dan sedang dikembangkan versi 6x. Untuk lebih lengkapnya dapat dilihat di situs resmi MySQL1.

Fitur-fitur MySQL antara lain :

- a. **Relational Database System.** Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
- b. **Arsitektur Client-Server.** MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada dikomputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
- c. **Mengenal perintah SQL standar.** SQL (Structured Query Language) merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL:2003.

- d. Mendukung **Sub Select**. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
- e. Mendukung **Views**. MySQL mendukung views sejak versi 5.0
- f. Mendukung **Stored Prosedured (SP)**. MySQL mendukung SP sejak versi 5.0
- g. Mendukung **Triggers**. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.
- h. Mendukung **replication**.
- i. Mendukung transaksi.
- j. Mendukung **foreign key**.
- k. Tersedia fungsi GIS.
- l. Free (bebas didownload)
- m. Stabil dan tangguh
- n. Fleksibel dengan berbagai pemrograman
- o. Security yang baik
- p. Dukungan dari banyak komunitas
- q. Perkembangan software yang cukup cepat.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-

proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

2.10 Cost and Benefit

Analisa Cost and Benefit pada perhitungan investasi pengembangan teknologi informasi menggunakan prinsip perbandingan biaya yang harus dikeluarkan dengan manfaat yang dikeluarkan oleh perusahaan. Pendekatan ini biasanya digunakan pada saat situasi dimana penggunaan teknologi informasi memberikan mafaat yang *tangible* dan cenderung mudah diukur (*measurable*) secara kuantitatif. Ada beberapa metode yang digunakan dalam melakukan pendekatan perhitungan Cost and Benefit terkait dengan memanfaatkan IT. Diantaranya adalah :

1. Cost Displacement yaitu mereduksi biaya yang harus dikeluarkan oleh perusahaan.
2. Cost Avoidance yaitu menghindari biaya yang harus dikeluarkan oleh perusahaan.
3. Decision Analys yaitu memperbaiki kualitas keputusan yang diambil.
4. Impact Analys yaitu menghasilkan dampak positif yang diperoleh oleh perusahaan.

Dalam penulisan tugas akhir ini penulis menggunakan metode Cost Displacement karena dianggap mudah dan paling tepat dalam merumuskan cost and benefit dalam pengembangan teknologi informasi.

a. Cost Cisplacement

Banyak Biaya yang dapat direduksi dengan dimanfaatkannya komputer atau teknologi informasi di sebuah perusahaan.

Biaya Investasi		Manfaat Bulanan	
Personal Computer	Rp.100	Reduksi Gaji Pegawai	Rp.42
Aplikasi Spreadsheet	Rp.128	Reduksi Proses Control	Rp.8
Jaringan	Rp. 73	Reduksi Biaya Admin	Rp. 4
Modem	Rp. 2	Reduksi Biaya Sewa	Rp. 2
Printer dan Scaner	Rp. 2	Reduksi Biaya Lain2	Rp. 1
Instalasi	Rp. 10	Total	Rp.57
Total	Rp.315		

Biaya Bulanan		Keuntungan Perbulan	
Karyawan	Rp. 9		Rp. 12
Pemeliharaan	Rp. 12	Manfaat Per Tahun	Rp.144
Pengembangan	Rp. 8	ROI	45.71 %
Lain- Lain	Rp.8	Simple Payback	2 Tahun
Amortisasi	Rp. 8		
Total	Rp. 45		

b. ROI dan Simple Payback

Keuntungan per Bulan = \sum Manfaat Bulanan - \sum Biaya Bulanan

Manfaat Per tahun - Keuntungan Per Bulan x 12 Bulan

$$\text{ROI} = \frac{\text{Manfaat Per Tahun}}{\sum \text{Biaya Investasi}} \times 100\%$$

Jika manfaat per tahun yang diperoleh besarnya tetap, Maka

$$\text{Simple Payback} = \frac{\sum \text{Biaya Investasi}}{\text{Manfaat Per tahun}} \times 1 \text{ Tahun}$$

c. Discounted Annual Net Benefit dan Discounted Payback

$$\text{Discounted Annual Net Benefit} = \frac{\text{Manfaat Per Tahun}}{(1+i)^t}$$

Jika Manfaat Per tahun yang diperoleh dihasilkan tetap maka

$$\text{Discounted Payback} = \frac{\sum \text{Biaya Investasi}}{\text{Discounted Annual Net Benefit}} \times 1 \text{ Tahun}$$

