

APLIKASI PENYUSUN SPESIFIKASI *HARDWARE* KOMPUTER DENGAN PYTHON UNTUK MEMBANTU PENJUALAN KOMPUTER

Rendy Adrianto L2F301492

Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang

Abstraksi

Toko komputer merupakan salah satu bisnis yang berkembang dengan perkembangan pesat di bidang hardware maupun software.

Pembuatan daftar spesifikasi komputer sesuai kebutuhan pelanggan, terkadang memerlukan waktu yang relatif lama. Hal ini dikarenakan beragamnya jenis hardware komputer untuk berbagai kebutuhan dan harga yang selalu berubah. Faktor kesalahan manusia (*Human Error*) juga dapat membuat hasil yang kurang konsisten. Untuk itu dirasa perlu adanya sebuah alat bantu yang dapat menjawab kebutuhan pelanggan atas fungsi komputer yang diinginkan beserta harganya dengan cepat dan akurat.

Python merupakan bahasa pemrograman yang menawarkan kemudahan dalam menuliskan suatu program dan dapat beroperasi hampir di semua platform. Python banyak diminati karena kesederhanaannya, dan dibalik kesederhanaannya python mendukung banyak pustaka yang tersimpan dalam modul-modul terpisah yang dapat digunakan kapanpun dibutuhkan. Python dapat digunakan untuk berbagai jenis aplikasi seperti aplikasi perkantoran, aplikasi web, dan lain-lain.

Dalam makalah Tugas Akhir ini dibahas secara singkat sebuah aplikasi untuk konsultasi spesifikasi hardware komputer untuk membantu penjualan pada toko komputer berdasarkan kebutuhan dan kelas nya dengan menggunakan bahasa pemrograman Python.

I. PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan komponen – komponen perangkat keras (*hardware*) komputer pada era teknologi informasi saat ini sangat pesat dan makin beragam dengan munculnya perangkat keras dengan berbagai jenis dan merk baru dengan kelebihan dan kekurangan masing-masing. Hal ini juga disertai dengan harga hardware yang bervariasi dan selalu berubah. Sebagai akibatnya, masyarakat pengguna komputer tidak mudah untuk dapat mengikuti perkembangan kemajuan teknologi komputer yang ada saat ini.

Sebuah toko komputer, tempat pelanggan mendapatkan komputer, biasanya menemui hambatan dalam menentukan spesifikasi *hardware* komputer yang diinginkan, karena kebutuhan yang beragam dan harga yang bervariasi serta sering berubah. Faktor

kesalahan manusia (*Human Error*) sangat berpengaruh terhadap keakuratan dan konsistensi dari hasil yang dihasilkan. Oleh karena itu dirasa perlu untuk membuat program aplikasi yang dapat membantu pekerjaan bagian penjualan untuk dapat menghasilkan spesifikasi komputer yang dibutuhkan pelanggan dengan cepat, akurat dan fleksibel.

1.2. Pembatasan Masalah

- Aplikasi yang dibuat di dalam Tugas Akhir ini menggunakan bahasa pemrograman Python 2.3.3.
- Sistem yang dibuat hanya berfungsi untuk menghasilkan *output* spesifikasi *hardware* komputer rakitan untuk kebutuhan :
 - PC untuk perkantoran,
 - PC untuk *Home Theater*,
 - PC untuk *Gaming* (permainan)
 - PC untuk *Video Editing* dan desain grafis
 - PC untuk anak-anak,
 - PC untuk komputer klien Warung Internet
 - PC untuk komputer server Warung Internet
- Aplikasi ini tidak melibatkan transaksi jual-beli komputer dengan konsumen, pemesanan, pembayaran maupun masalah administrasi lainnya.
- Sistem ini hanya diterapkan untuk pembelian komputer rakitan dan merupakan komputer yang baru.
- Hardware-hardware* yang direkomendasikan disini merupakan *hardware-hardware* yang beredar sampai dengan tahun 2005.
- Solusi yang dihasilkan oleh sistem ini hanya bersifat rekomendasi, bukan merupakan solusi yang mutlak.
- Sistem operasi yang digunakan dari keluarga *Microsoft Windows* dengan program *browser* standar *Microsoft Windows* yaitu *Internet Explorer*. Aplikasi ini tidak dibahas mengenai penggunaannya di dalam *platform* sistem operasi lain

II. LANDASAN TEORI

2.1. Basis Data

2.1.1. Pengertian Basis Data

Basis data adalah kumpulan data atau informasi yang teratur dan terkomputerisasi yang terdiri dari satu atau lebih tabel yang saling berhubungan. Kumpulan data tersebut bersifat

minimum *redundancy* (kerangkapan data). Basis data melayani banyak pengguna secara optimal. Basis data dapat dikategorikan sangat spesial dalam dunia komputer karena menjadi hal yang utama dalam analisa dan perancangan sistem.

2.2. Python

Python merupakan bahasa pemrograman yang menawarkan kemudahan dalam menuliskan suatu program dan dengan bahasa yang *multiplatform* dapat berjalan dalam lingkungan Windows, UNIX, Linux dan Mac dengan memberikan probabilitas yang tinggi^[3]. *Python* banyak diminati karena kesederhanaannya, dan dibalik kesederhanaannya *python* mendukung banyak pustaka yang tersimpan dalam modul-modul. Sejumlah pustaka yang tersedia antara lain mendukung untuk jaringan, antarmuka, grafis, pencitraan, analisis dan komputasi numerik, *hypertext* (**HTML**,**XML**, dll), akses database, dan berbagai hal lain. Hal-hal yang perlu diperhatikan dalam Python :

a. Membuat Skrip (*Scripting*)

Biasanya dalam membuat satu program yang kompleks adakalanya kumpulan instruksi dijadikan satu dalam satu berkas, terutama apabila sering menggunakan sekumpulan intruksi untuk melaksanakan satu tugas yang sifatnya rutin. Skrip *python* ditulis dengan akhiran *.py*

b. Modul

Setiap Skrip pada *python* dianggap sebagai modul. Intinya Modul adalah kode yang disimpan dalam sebuah berkas dalam media penyimpanan eksternal. Selanjutnya, kode yang terdapat dalam modul dapat dipergunakan dalam suatu skrip dengan terlebih dahulu mengimpor (mengambil) modul tersebut.

c. Objek

Di dalam *python*, objek merupakan abstraksi terhadap data. Setiap data dinyatakan dalam objek. Sebuah objek memiliki nilai dan tipe, misalnya ketika diberikan perintah `a=5`, maka sebenarnya `a` adalah objek. Tipenya adalah bilangan bulat dan nilainya adalah 5. Dan yang perlu diketahui begitu diberikan nilai pada suatu objek maka identitas objek tersebut tidak berubah.

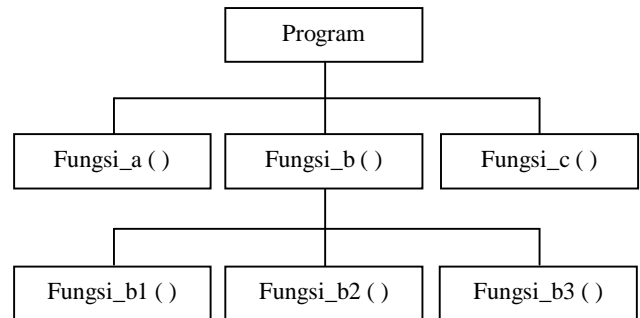
d. Identasi

Identasi (penulisan text yang menjorok ke kanan) memegang peranan penting dalam penulisan pernyataan pada *python* mengingat *identasi* digunakan sebagai blok kode pada pernyataan seperti *if*, *while*, dan *for*.

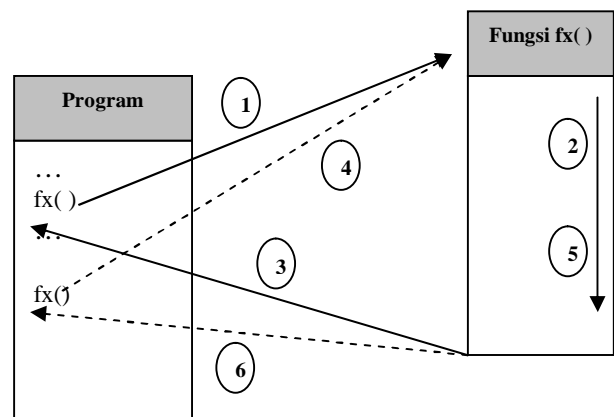
e. Fungsi

Sebuah program yang besar biasanya tersusun atas sejumlah fungsi. Sebuah fungsi berisi

sejumlah pernyataan yang dikemas dengan sebuah nama. Selanjutnya nama ini dapat dipanggil beberapa kali dalam program dan tentu saja cara ini dapat mengurangi duplikasi kode. Alasan lain pembuatan fungsi adalah untuk membuat suatu program agar dapat dipecah menjadi sejumlah bagian agar dapat dikelola dengan lebih mudah oleh pemrogram daripada kalau hanya berupa satu bagian kode yang besar.



Gambar 2.1 Program tersusun atas sejumlah fungsi



Gambar 2.2 : Sebuah fungsi dapat dipanggil beberapa kali di beberapa tempat. Urutan angka menyatakan urutan proses

f. Namespace

Yang dimaksud dengan *namespace* adalah satu pemetaan antara nama (ataupun pengenal) terhadap objek. Dengan menggunakan *namespace* inilah *python* dapat mengakses nama-nama. Wujud *namespace* adalah seperti kamus. Di dalamnya tersimpan kata-kunci (yaitu berupa nama) dan nilainya (nilai dari nama).

g. Struktur Data

Yang dimaksud dengan struktur data pada *python* adalah suatu objek yang dapat menyimpan sejumlah data. Termasuk dalam struktur data yaitu *list*, *tuple* dan *dictionary*.

- *List* merupakan suatu bentuk struktur data yang bisa mengandung sejumlah elemen dengan tipe yang berbeda-beda dan memungkinkan penambahan atau pengurangan elemen-elemennya.
- *Tuple*, yang hampir menyerupai *list*, namun struktur data ini bersifat *immutable* (bagian isinya tidak bisa dirubah).
- *Dictionary*, atau kamus. Di dalam kamus biasanya yang dicari adalah arti dari suatu kata kunci. Dalam *dictionary* ini hal serupa juga dilakukan yaitu menggunakan kata kunci untuk mendapatkan suatu nilai.

h. Kesalahan Dan Eksepsi

Dunia pemrograman tidak dapat lepas dari kesalahan. Kesalahan ini dapat di klasifikasikan dalam 3 hal:

- Kesalahan sintaksis, yakni kesalahan yang disebabkan pernyataan ditulis tidak mengikuti kaidah yang benar.
- Kesalahan sewaktu eksekusi, yang mengakibatkan eksekusi segera dihentikan.
- Kesalahan logika, yakni kesalahan yang terdeteksi sewaktu skrip dieksekusi tetapi tidak menghentikan eksekusi skrip.

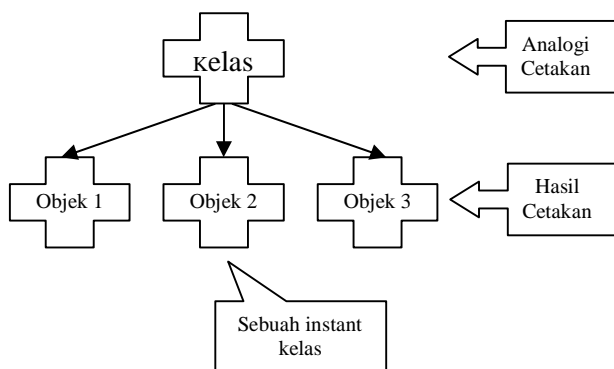
Pada *python*, kesalahan sintaksis dan kesalahan sewaktu eksekusi akan menimbulkan *eksepsi*. Dalam hal ini eksepsi dapat dipandang sebagai suatu tindakan yang mengisaratkan satu kesalahan sewaktu eksekusi terjadi. Normalnya *eksepsi* akan membuat interpreter *python* menampilkan satu pesan kesalahan dan kemudian menghentikan eksekusi terhadap pernyataan atau skrip. Aneka *eksepsi* pada *python* dapat dilihat dengan menuliskan perintah seperti berikut pada interpreter *python*:

```
>>> dir (__builtins__)
```

Nama-nama *eksepsi* umumnya ditandai dengan kata **error** misalnya **IOError** dan **KeyError**. Akan tetapi tidak semua hasil yang diberikan oleh pernyataan diatas adalah nama-nama *eksepsi*.

i. Kelas Dan Pewarisan

Di dalam terminologi pemrograman berorientasi objek, suatu objek dibentuk oleh suatu kelas. Itulah sebabnya kelas dapat dikatakan sebagai cetakan untuk membuat objek-objek. Objek yang dibentuk melalui kelas dinamakan *instant kelas*.



Gambar 2.3 Kelas sebagai cetakan objek-objek

Kelas memungkinkan pemrograman dapat membuat sendiri suatu tipe yang belum dikenal sebagai satu tipe yang didefinisikan pemakai (*user defined typed*)

Seperti halnya fungsi, kelas harus didefinisikan terlebih dahulu sebelum dapat dipakai untuk menentukan objek.

Sebuah kelas tidak hanya dapat mengandung makna atribut, tetapi bisa mengandung metode. Sebagaimana diketahui, metode adalah suatu fungsi yang melekat pada objek.

Dalam terminologi pemrograman berorientasi objek, terdapat istilah konstruktor dan destruktur. Konstruktor adalah metode yang akan dijalankan ketika suatu objek diciptakan, sedangkan destruktur adalah metode yang akan dijalankan ketika objek dimusnahkan. Di dalam *python*, konstruktor berupa metode dengan nama `__init__()`, sedangkan destruktur berupa metode dengan nama `__del__()`. Semua metode (termasuk destruktur dan konstruktor) dalam kelas selalu mengandung argument pertama berupa kata **self**, yang menyatakan objek bersangkutan.

2.3. ZODB (Z OBJECT DATA BASE)

ZODB (*Z Object Database*) merupakan bahasa pemrograman lanjutan yang menyediakan fasilitas-fasilitas yang secara otomatis menuliskan objek pada media penyimpanan (misalnya :disk) dan dapat membaca objek tersebut ketika dibutuhkan pada saat program dijalankan. Dengan kata lain dengan menginstal ZODB maka secara otomatis fasilitas tambahan ini akan masuk kedalam *library python*. Kemudahan dari ZODB selain terintegrasi dengan *python*, ZODB memudahkan administrator untuk mengatur objek secara mudah. Seperti halnya *software database* yang lain ZODB dapat diakses dan dirubah menurut keinginan, dan proses seperti ini dinamakan *transaction*. ZODB memiliki kelebihan fasilitas yang disebut ACID (*Atomicity, Consistency, Isolation and Durability*). Fasilitas ini adalah fasilitas yang mampu mempertahankan suatu data dalam satu database agar tidak rusak (*corrupt*) ketika terjadi *crash* pada *hardware* maupun *software*.

1. Atomicity

Atomicity berarti perubahan apapun yang dilakukan pada data dalam suatu proses *transaction* / transaksi tidak akan berpengaruh apapun pada data. Maksudnya adalah jika suatu program menjalankan beberapa perubahan data dan kemudian terjadi *crash*, maka database tidak akan berubah ketika meninggalkan data dalam keadaan tidak stabil.

2. Consistency

Consistency berarti bahwa masing-masing transaksi melakukan satu perubahan secara valid pada suatu database. Misalnya konsistensi dalam *chek-in* database atau bahasa.

3. Isolation

Isolation dapat berarti penyekatan. Maksudnya adalah jika ada 2 program berjalan dalam 2 transaksi yang berbeda, masing-masing dari mereka tidak dapat melihat perubahan satu dengan yang lain sampai mereka menyetujui transaksinya.

4. Durability

Durability atau biasanya disebut dengan ketahanan bermakna bahwa jika transaksi sudah disetujui maka ketika terjadi *crash* tidak akan membuat data menjadi hilang atau *corrupt*.

Dalam pemrograman ZODB, ada 3 macam *interface* yang mendukung diantaranya *Storage*, *DB*, *Connection*. Secara berurutan dalam menjalankan ZODB pertama kali yang harus dilakukan adalah membuka *Storage* (biasanya berupa file yang bernama *FileStorage*, *DBFullStorage*, *DemoStorage* *dsb*), kemudian membuat *DB* yang akan menghubungkan dengan *Connections*.

ZODB dipakai karena tidak perlu banyak menulis kode untuk memanggil atau memasukkan objek dari atau ke database. Tinggal meletakkan objek *persistent* (terus menerus bisa berubah) yang ingin dipakai saja ke dalam suatu folder yang akan dipakai, misalnya saja pada *dictionary python*. Semua yang terdapat dalam *dictionary* ini akan disimpan dalam database. Bisa dikatakan bahwa *dictionary* adalah "root" dari database. Objek *python* apapun yang diletakkan di dalamnya akan menjadi *persistent*

Karena program ini terintegrasi dengan *python* maka ZODB dapat dijalankan tanpa harus menjalankan skripnya dahulu untuk membuat database. Jika tak ada database yang terdeteksi maka secara otomatis program akan menciptakannya sendiri dan mengenalinya.

III PERANCANGAN APLIKASI PENYUSUN SPESIFIKASI HARDWARE KOMPUTER

3.1. Rekayasa Sistem

Elemen – elemen yang dibutuhkan dalam aplikasi ini adalah sebagai berikut

1. Data

Basis data merupakan kumpulan data *non-redundant* yang dapat digunakan bersama oleh sistem – sistem aplikasi yang berbeda. Dengan kata lain, basis data adalah kumpulan data-data yang saling terkait satu sama lainnya dalam membentuk bangunan informasi yang penting.

Langkah – langkah membuat aplikasi ini

secara umum dimulai dengan mencari data tentang pemilihan spesifikasi komputer dari buku dan majalah, lalu mengkonsultasikan pada pakar. Langkah lain yang dilakukan adalah mencari data tentang *hardware – hardware* yang sering dipilih *user* sebagai dasar pembuatan basis pengetahuan dan basis aturan.

Data yang dibutuhkan dalam rancangan sistem pakar ini adalah data keperluan penggunaan PC, kelas, nama *hardware* komputer, harga *hardware* komputer, kurs rupiah terhadap dollar, dan data basis pengetahuan dan aturan. Dari data tersebut, untuk mendapatkan informasi yang lebih detil dan terperinci, maka aplikasi ini harus mampu mengolah agar pemakai dapat dengan mudah mendapatkan informasi yang diinginkan. Hasil pengolahan data tersebut akan disajikan sebagai informasi spesifikasi komputer rakitan yang direkomendasikan aplikasi penyusun spesifikasi *hardware* ini beserta harganya.

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem yang akan dibuat adalah :

- Python-2.3.3*, memungkinkan pengembangan program antarmuka pemakai secara mudah dengan sedikit pengkodean sehingga memerlukan waktu yang singkat.
- ZODB*, digunakan untuk membuat basis data yang mudah dibangun dan dimanipulasi sehingga dapat dikoneksikan dengan antarmuka *Python-2.3.3*.
- Adobe Acrobat Reader*, versi 4.0 keatas, digunakan untuk menampilkan hasil penyusunan spesifikasi ke dalam bentuk file *.pdf.

3. Perangkat Keras

Perangkat keras yang mendukung sistem pakar ini tidak harus berspesifikasi tinggi. Namun direkomendasikan untuk menggunakan spesifikasi perangkat keras sebagai berikut :

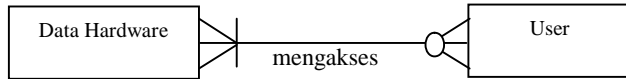
- CPU*. Kebutuhan yang harus dipenuhi adalah menggunakan dengan CPU yang memiliki prosesor berkemampuan tinggi seperti *Intel Pentium III* atau *IV*.
- RAM*. Direkomendasikan menggunakan RAM minimum 64 Mb. Kondisi optimal adalah dengan menggunakan RAM 256 Mb.
- Hardisk*. Data yang akan dimasukkan ke dalam *hardisk* adalah data – data *hardware* komputer yang sederhana sehingga tidak perlu menggunakan *hardisk* yang besar. *Hardisk* 20 GB sudah cukup.
- Input device*. Perangkat ini merupakan peralatan-peralatan yang digunakan untuk memasukkan data ke dalam sistem. *Keyboard* dan *mouse* sudah cukup.
- Output device*. Perangkat ini merupakan peralatan-peralatan yang digunakan untuk

menampilkan data dan informasi. Yang termasuk dalam perangkat ini adalah *monitor*, dan *printer*.

3.2. Tahapan Perancangan

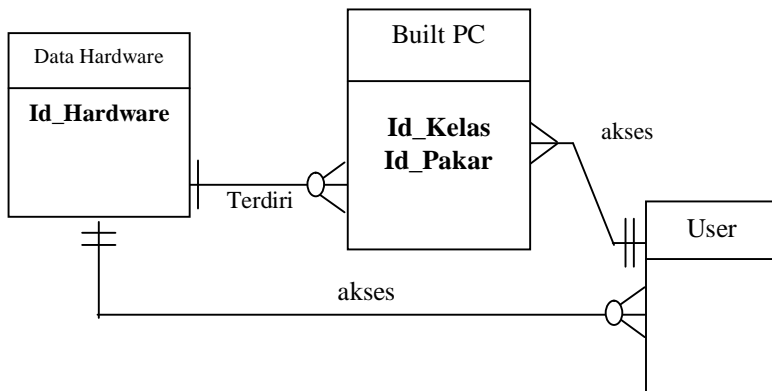
3.2.1. ERD (*Entity Relationship Diagram*)

Langkah pertama yang dilakukan adalah menentukan entitas, kemudian terlebih dahulu dibuat model data konteks atau diagram struktur data dari entitas sistem yang telah ditentukan. Pada tugas akhir ini ditunjukkan oleh gambar 3.4 sebagai berikut:



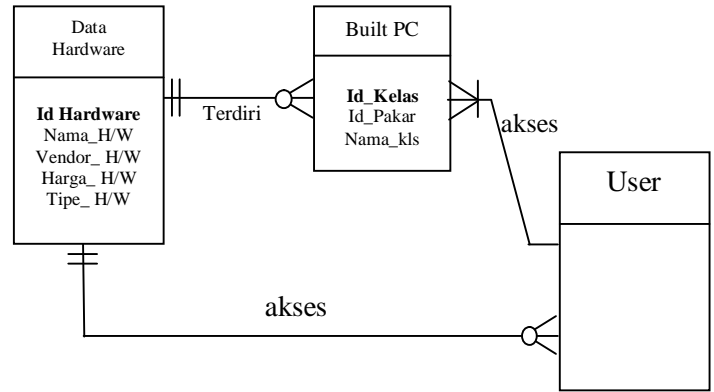
Gambar 3.1 Model data konteks aplikasi penyusun spesifikasi komputer pada toko komputer

Setelah hubungan entitas didefinisikan, kemudian menterjemahkannya kedalam satu tabel dengan terlebih dahulu menentukan atribut kunci masing-masing entitas. Penentuan kunci ini menjadi penting dalam basis data relasional karena digunakan untuk mengakses suatu baris tertentu. Model data berdasarkan kunci primer untuk sistem *e-commerce* ditunjukkan oleh gambar 3.5.



Gambar 3.2 Model data berdasarkan kunci pada aplikasi penyusun spesifikasi komputer pada toko komputer

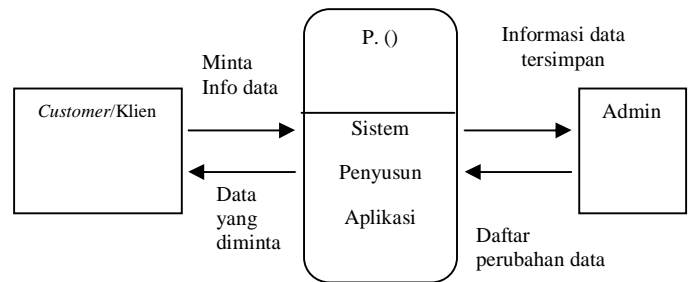
Selanjutnya dibuat model data yang menggambarkan entitas lengkap dengan seluruh atributnya yang akan menjadi acuan dalam membangun table-table basis data. Entitas akan diterjemahkan menjadi nama susunan tertentu dan atribut dari suatu entitas akan menjadi nama-nama kolom dari table tersebut. Model data tersebut ditunjukkan oleh gambar 3.3 sebagai berikut



Gambar 3.3 Model data lengkap dengan atribut pada aplikasi penyusun spesifikasi komputer pada toko komputer

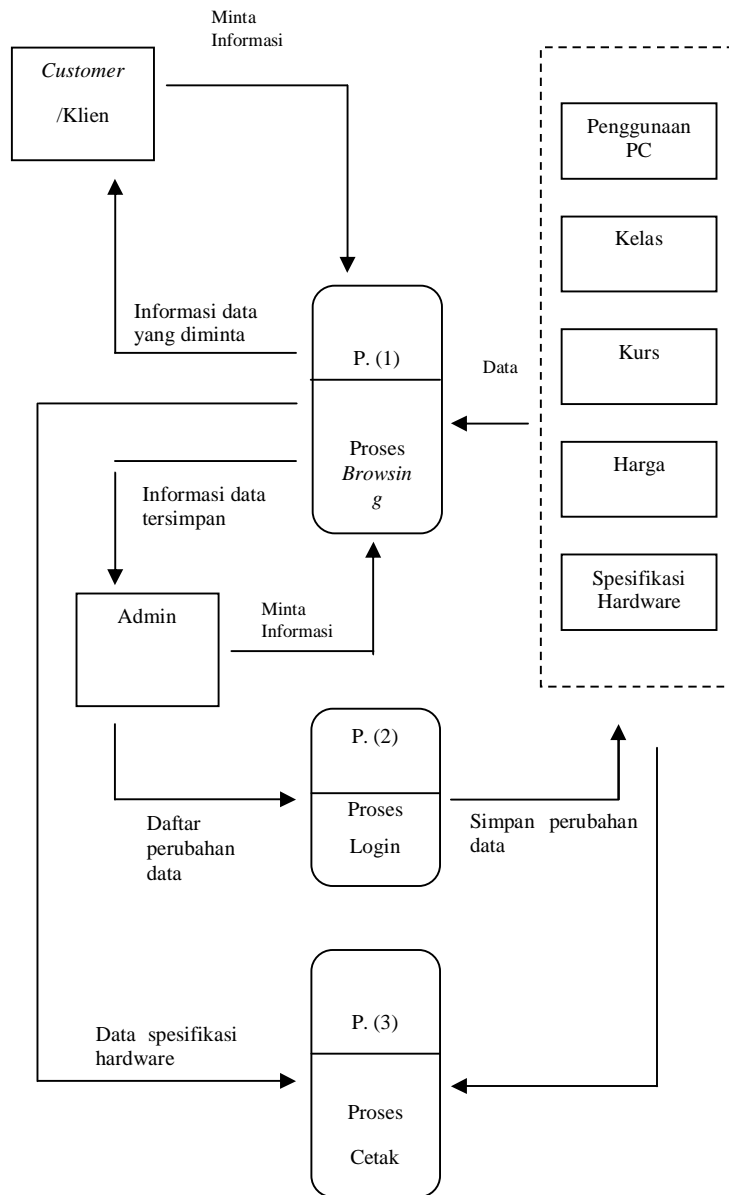
3.2.2. DFD (*Data Flow Diagram*) Aplikasi Penyusun Spesifikasi Hardware

Untuk menentukan ruang lingkup sistem yang akan dibangun, sesuai dengan kebutuhan sistem yang telah dianalisa sebelumnya, dijabarkan dalam bentuk diagram konteks seperti pada gambar 3.4.



Gambar 3.4 Diagram konteks aplikasi penyusun spesifikasi komputer pada toko komputer

Selanjutnya dari diagram konteks, dijabarkan ke dalam menjadi Data Flow Diagram (DFD) untuk mendapatkan gambaran sistem yang lebih detail. Gambar 3.5 memperlihatkan DFD dan aplikasi penyusun spesifikasi komputer pada toko computer yang merupakan penjabaran dan diagram konteks sistem.

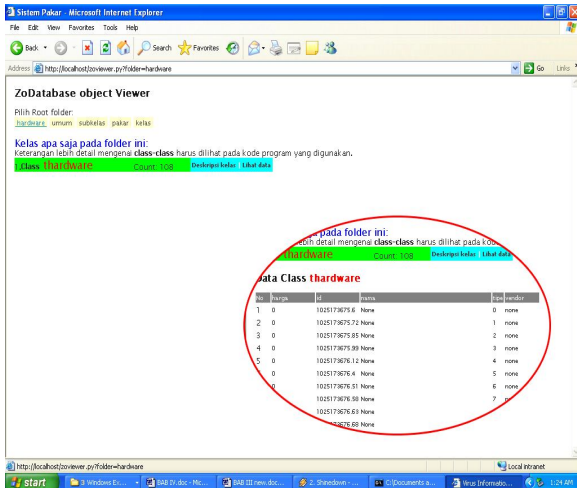


Gambar 3.5 DFD Level 0 aplikasi penyusun spesifikasi komputer pada toko komputer

IV IMPLEMENTASI DAN PENGUJIAN PROGRAM

4.1. Implementasi Database

Database pada sistem toko komputer yang menggunakan *e-commerce* ini, dibuat menggunakan database ZODB. Dalam proses pembuatannya database ini disimpan dengan nama *sistempakar.py*. Jika database ini dapat ditampilkan dalam bentuk HTML maka hasilnya akan tampak sebagai berikut :

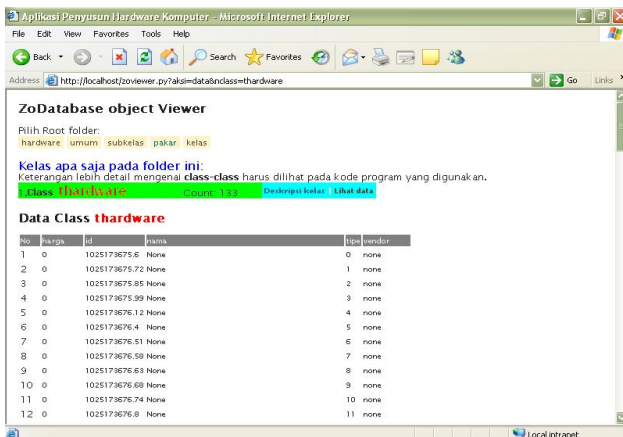


Gambar 4.1 Tampilan ZODB viewer "Toko komputer" menggunakan HTML

4.2. Pengkodean Program

1. Hardware

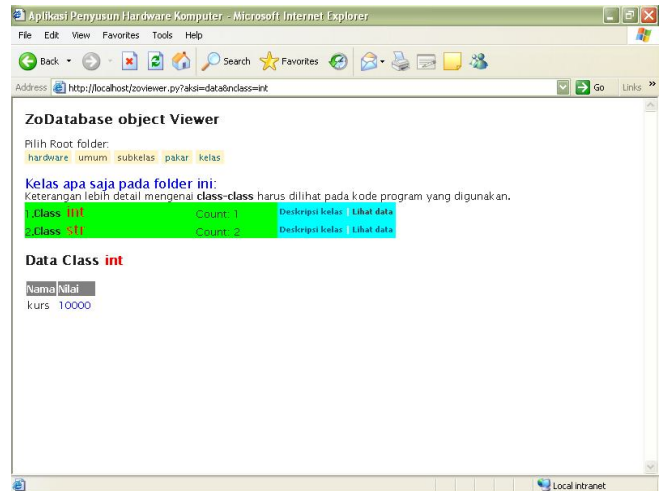
Pada folder hardware ini menampilkan kelas – kelas yang terdiri atas vendor, id, nama, harga dan type. Kelas-kelas hardware ini berfungsi untuk menampilkan data-data mengenai vendor sebuah hardware komputer dengan tipe dan harga tertentu. Jika diperlihatkan dalam ZODB viewer akan tampak seperti Gambar 4.2.



Gambar 4.2 Tampilan ZODB Viewer "Toko komputer" hardware

2. Umum

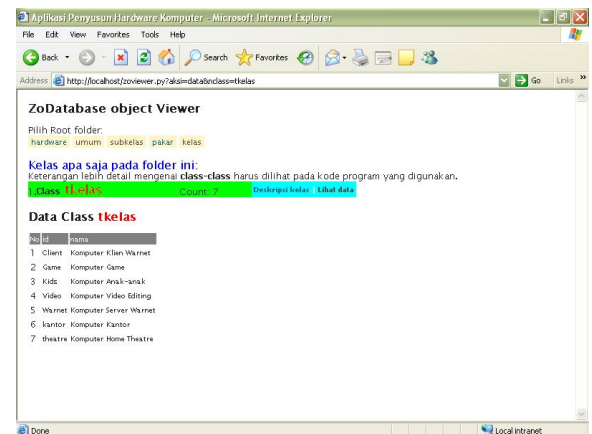
Pada folder umum ini berisi tentang *class* yang digunakan untuk menyimpan perubahan password pada admin, waktu update hardware dan juga menentukan nilai kurs dolar terhadap rupiah. Tampilan ZODB Viewernya sebagai berikut:



Gambar 4.3 Tampilan ZODB Viewer "Toko komputer" umum

3. Kelas

Kelas ini adalah bagian yang paling sederhana dari database yang dibuat, kelas ini berfungsi untuk menampung beberapa nama klasifikasi komputer yang ingin dikelompokkan beserta dengan kode dan deskripsi (nama). Tampilan ZODB Viewernya sebagai berikut:

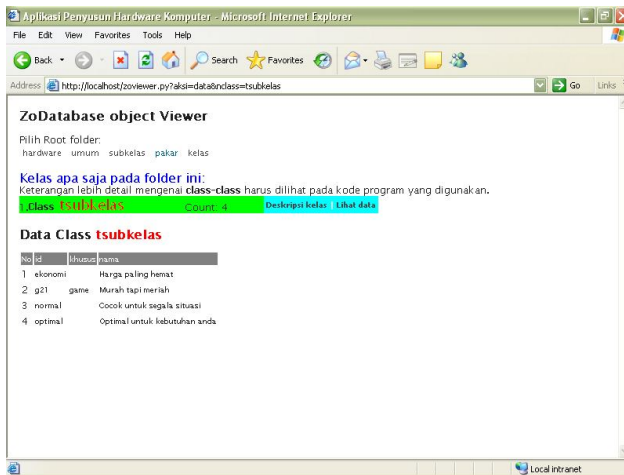


Gambar 4.4 Tampilan ZODB Viewer "Toko komputer" kelas

4. Subkelas

Subkelas sebenarnya mengacu kepada suatu folder yang digunakan untuk menampung beberapa pilihan spesifikasi komputer yang telah ditentukan oleh admin, seperti halnya kelas. Subkelas ini adalah bagian dari kelas yang dibuat lebih spesifik lagi untuk memudahkan seorang klien memilih sesuai dengan

kebutuhan. Tampilan ZODB Viewernya sebagai berikut:



Gambar 4.5 Tampilan ZODB Viewer "Toko komputer" subkelas

a. Pengujian Program

1. Halaman Index

Halaman index seringkali dikatakan sebagai halaman depan. Halaman ini merupakan halaman utama aplikasi ini. Dalam halaman ini seorang user dihadapkan dengan 3 pilihan, yaitu menuju halaman admin, halaman sispakar dan halaman hardware.



Gambar 4.6 Tampilan halaman index "Toko komputer" pada HTML

2. Halaman Admin

Halaman ini berfungsi untuk mengubah nilai dan semua parameter yang ada, dengan cara *login* terlebih dahulu menggunakan *password*.



Gambar 4.7 Tampilan halaman admin "Toko komputer" pada HTML

Jika password yang diketikkan benar maka nilai kurs rupiah, paket harga, jenis barang dll dapat diubah sewaktu-waktu mengingat harga komputer tergantung nilai dolar yang berubah tiap hari dan juga model hardware yang bermunculan semakin variatif.



Gambar 4.8 Tampilan halaman admin2 "Toko komputer" pada HTML

3. Halaman Sispakar

Halaman sispakar berisi tentang spesifikasi hardware yang dibutuhkan. Halaman ini akan memperlihatkan beberapa spesifikasi hardware komputer yang telah dirancang oleh admin sesuai dengan kebutuhan, misalnya untuk keperluan kantor dengan harga yang paling hemat maka akan tampak seperti gambar 4.9

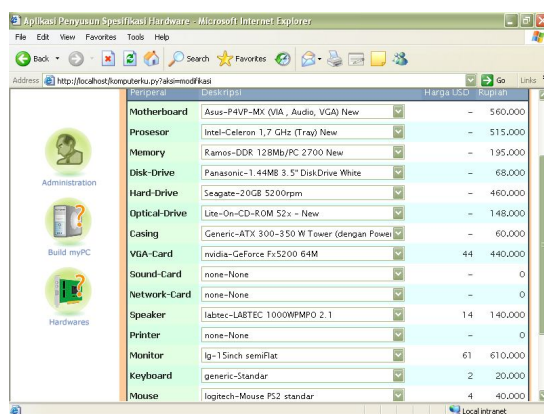


Gambar 4.9 Tampilan halaman sispakar1 "Toko komputer" pada HTML



Gambar 4.10 Tampilan halaman sispakar2 "Toko komputer" pada HTML

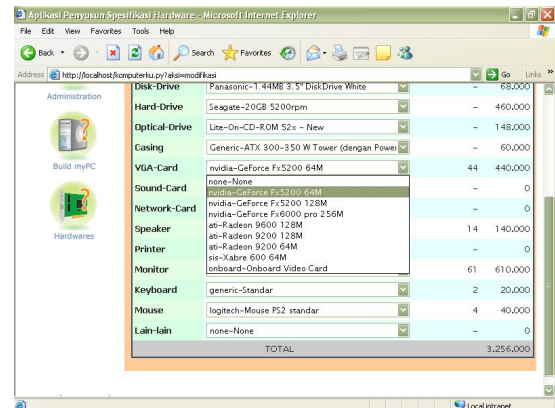
Setelah itu akan ditampilkan halaman HTML yang menampilkan spesifikasi hardware yang diminta dengan harga hemat



Gambar 4.11 Tampilan halaman sispakar3 "Toko komputer" pada HTML

Kelebihan lain dari halaman ini adalah setelah muncul suatu *result* dari spesifikasi hardware ternyata hasil yang telah diminta dapat diubah sesuai

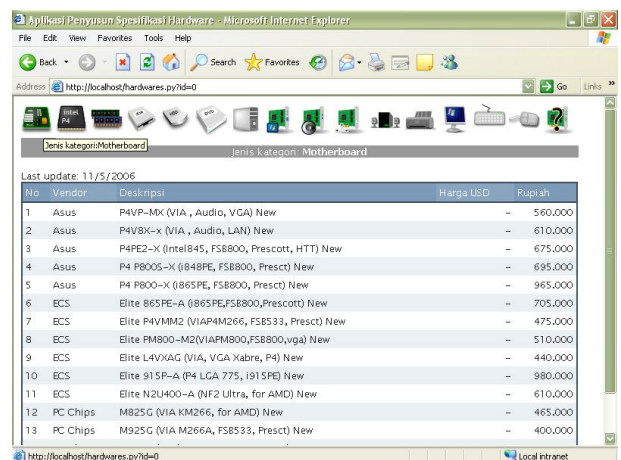
dengan keinginan baik dengan menambah hardware atau mengurangi hardware sesuai dengan kebutuhan dan dana pengguna. Hasilnya akan tampak pada gambar 4.12 yang dapat diupdate (berfungsi untuk menampilkan tampilan modifikasi baru beserta dengan perubahan harga) dan distore (berfungsi untuk menampilkan tampilan solusi baru yang disimpan sementara dan dapat dilihat sewaktu waktu untuk dibandingkan dengan solusi spesifikasi yang lain).



Gambar 4.12 Tampilan halaman sispakar4 "Toko komputer" pada HTML

3. Halaman Hardware

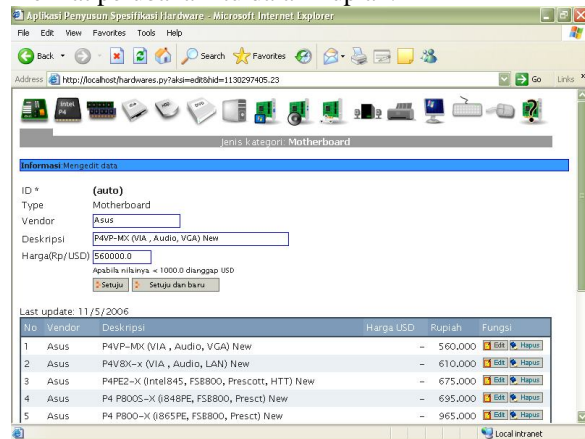
Halaman hardware ini berisi tentang spesifikasi hardware secara detail, yang dapat diakses dengan mudah dengan berbagai pilihan mulai dari spesifikasi VGA card, SoundCard, Motherboard, Processor dll dengan berbagai merk dan spesifikasi dengan disertai dengan harga.



Gambar 4.13 Tampilan halaman hardware "Toko Komputer" pada HTML

Dari sisi klien, klasifikasi hardware beserta harga hanya dapat dilihat, dipilih dan dicetak saja. Akan tetapi bagi admin dapat mengadakan perubahan total dengan mengedit atau menghapusnya. Bahkan jika perlu dapat menambahkan vendor dari tipe hardware beserta harganya tanpa perlu mengkonversi

harga dalam bentuk rupiah, karena secara otomatis jika memasukkan nilai harga dalam bentuk dolar maka secara otomatis seorang klien akan dapat melihat perubahan itu dalam rupiah.



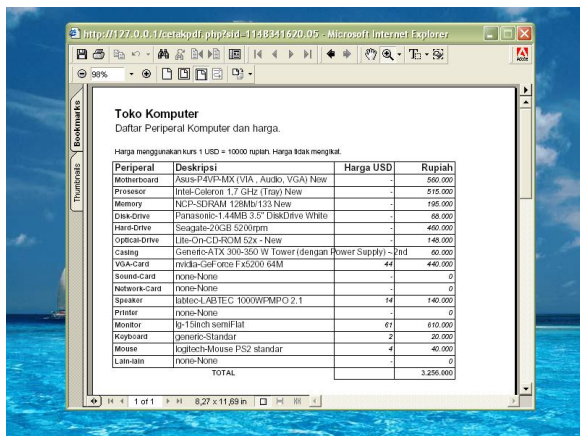
Gambar 4.14 Tampilan halaman hardware2 "Toko Komputer" pada HTML

5. Hasil Cetak

Hasil Cetak yang dihasilkan oleh aplikasi ini ada dua jenis yang ditampilkan oleh program *Acrobat Reader*, yaitu :

- Hasil cetak spesifikasi komputer yang disarankan

Hasil cetak ini dihasilkan ketika pengguna menekan tombol "Cetak" pada halaman tersebut dan akan menghasilkan output sebagai berikut :

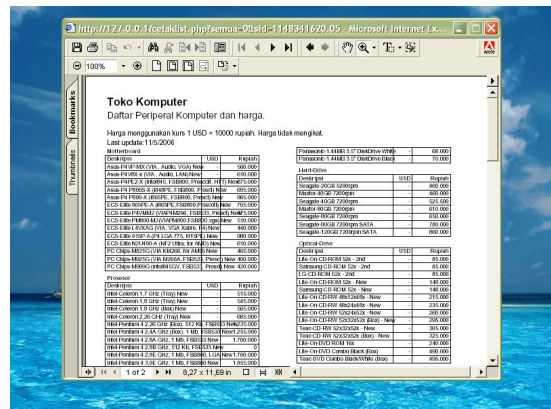


Gambar 4.15 Tampilan hasil cetak rekomendasi penyusunan spesifikasi yang dihasilkan

- Hasil Cetak daftar hardware yang dimiliki toko komputer secara keseluruhan sehingga membentuk daftar harga (*Price List*)

Hasil cetak ini dihasilkan ketika pengguna

menekan tombol "Cetak Price List" pada halaman tersebut dan akan menghasilkan output sebagai berikut :



Gambar 4.16 Tampilan hasil cetak keseluruhan daftar hardware yang dimiliki toko tersebut (*Price List*)

Untuk mencetak hasil tampilan ke dalam bentuk kertas, pengguna dapat menekan gambar printer sehingga printer yang terpasang pada komputer dapat mencetaknya ke dalam bentuk kertas. Sebelumnya komputer yang bersangkutan harus diinstal terlebih dahulu dengan sebuah printer yang siap untuk digunakan.

V. KESIMPULAN

5.1. Kesimpulan

Dari perancangan aplikasi yang telah dilakukan diperoleh kesimpulan bahwa :

- Aplikasi Penyusun Spesifikasi *Hardware* Komputer ini memberi kemudahan kepada pengguna dengan memberikan rekomendasi spesifikasi komputer rakitan dan harganya.
- Aplikasi Penyusun Spesifikasi *Hardware* Komputer dapat melakukan modifikasi solusi untuk menghasilkan solusi yang sesuai dengan selera konsumen.
- Aplikasi ini juga dapat memperlihatkan daftar *hardware* yang tersedia di toko komputer tersebut yang terkatagori menjadi beberapa jenis *hardware* dari mulai dari : *Motherboard*, *Prosesor*, *RAM (Memori)*, *Disk drive*, *Hard disk*, *Optical drive*, *Casing*, *VGA Card* (kartu grafis), *Sound Card* (kartu suara), *Network Card* (kartu Jaringan), *Speaker*, *Printer*, *Monitor*, *Keyboard*, *Mouse*, *Perangkat pendukung (Peripheral)* lain.
- Sistem ini dapat memperbarui (*update*) database sehingga dapat mengubah informasi dan harga hardware, mengubah kurs rupiah terhadap dollar, dan mengubah aturan masukkan terhadap sistem.

5.2. Saran

Dari perancangan aplikasi yang telah dilakukan, Penulis mengusulkan saran-saran untuk pengembangan Aplikasi Penyusun Spesifikasi *Hardware* Komputer ini sebagai berikut :

1. Aplikasi ini dapat dikembangkan lebih lanjut menjadi sebuah sistem yang berbasis web.
2. Untuk mendukung kinerja penjualan komputer secara maksimal pada toko komputer tersebut secara luas, sistem ini perlu dikembangkan dengan dukungan teknologi *e-commerce* dengan fasilitas penjualan perangkat komputer secara *online* lengkap dengan sistem pembayaran *online* yang aman.

DAFTAR PUSTAKA

- a. Hendri, 2003, *Cepat Mahir Python*, www.ilmukomputer.com.
- b. Kadir, Abdul, 1999, *Konsep dan Tuntunan Praktis Basis Data*, Yogyakarta : Penerbit Andi
- c. Kadir, Abdul, 2005, *Dasar Pemrograman Python*, Yogyakarta: Penerbit Andi.
- d. Kuchling, A.M. 2004, *ZODB/ZEO Programming Guide*, <http://www.zope.org/Wikis/ZODB/guide/index.html>.
- e. Noprianto, 2002, *Python & Pemrograman Linux*, Yogyakarta : Penerbit Andi
- f. Zope.org, 2003, *ZODB for Python Programmer*. <http://www.zope.org/Wikis/ZODB1.html>



BIOGRAFI

Rendy Adrianto, lahir di Jakarta, 5 Desember 1978, saat ini sedang dalam taraf menyelesaikan Tugas Akhir untuk meraih gelar Sarjana di Jurusan Teknik Elektro Universitas Diponegoro Semarang dengan konsentrasi Teknik Komputer dan Informatika.

Mengetahui/Mengesahkan,
Dosen Pembimbing

Pembimbing I

Pembimbing II

Ir. Kodrat I S, MT
NIP. 132 046 696

Agung Budi P. ST, MIT
NIP. 132 137 932