

210/0065
10 APR 2010 A

Komputerisasi untuk administrasi pada sebuah Apotik dengan menggunakan Rational Rose® 2000

Aris Budiarto

Jurusan Teknik otomasi Manufaktur dan Mekatronika

Politeknik Manufaktur Negeri Bandung

Abstrak

Sebuah apotik akan membuat sistem komputerisasi pada administrasi kegiatannya. Komputerisasi tersebut meliputi pencatatan setiap transaksi baik dari pembelian langsung atau dengan resep. Untuk yang melalui resep akan disorting terlebih dulu dari pasien anggota askes atau umum, juga dilakukan pencatatan terhadap nama pasien, umur, alamat, nama dokter, alamat dokter, spesialisasi dokter, nama obat, jumlah transaksi, tanggal resep dan juga tanggal transaksi. Dari sisi lain seorang member apotik (karyawan yang mempunyai hak akses ke sistem komputerisasi apotik) dapat melihat omset penjualan obat, jenis modus obat yang banyak terjual, dan profit apotik dalam suatu termin waktu, serta jumlah peserta askes dan efektifitasnya atau bahkan stok (cadangan) obat-obatan. Hal ini untuk kepentingan menejerial apotik (continuous improvement).

Keyword : hak akses, disorting, komputerisasi

1. Spesifikasi Sistem (User Requirement)

- Pada batas tertentu pelanggan dapat mengakses homepage apotik lewat internet maupun intranet yang disediakan pihak apotik.
- Beberapa informasi berkenaan dengan apotik yang tak bersifat rahasia dapat di-download dengan cepat, meskipun keadaan jaringan sedang sibuk.
- Khusus pembelian ataupun resep tidak dapat dilakukan transaksi melalui internet hal itu untuk kepentingan originalitas resep dan keselamatan pasien serta tanggung jawab pihak apotik.
- Data-data harus aman (secure), dan harus terenkripsi dengan baik.
- Sistem dapat mensortir jenis obat untuk alternatif transaksi terhadap pelanggan, misalnya obat generik, pembelian dalam jumlah yang besar, dan memberitahukan jenis obat askes atau bukan.

- Sesuatu berkenaan dengan kerahasiaan apotik dapat diakses oleh seseorang yang cocok password-nya, ini untuk selektifitas sistem dan keamanan sistem, namun memudahkan dalam memonitor kemajuan apotik.
- Setiap transaksi dengan pihak apotik selalu langsung berhubungan dan sekaligus merubah database dari sistem (real time).

2. Requirement Analysis

2.1. Project Planning

- Perencanaan proyek ini akan melibatkan berbagai resource dan problem yang cukup kompleks. Misalnya kompleksitas software akan semakin tinggi karena yang dibutuhkan adalah sistem yang real time, dimana setiap akses transaksi selalu melibatkan database yang terkini. Meskipun tidak ada transaksi yang langsung lewat network (selalu tangibel). Selain itu sistem keamanan yang tinggi perlu sistem sekuritas yang tinggi juga.
- Waktu implementasi dari perancangan ini minimal memakan waktu 1,5 bulan jika dikerjakan dengan efektif dan

memenuhi minimal spesifikasi (minimum requirement) sesuai standard level agreement (SLA). Dengan perincian 2 minggu waktu efektif untuk menangani database, 2 minggu efektif untuk mengerjakan jaringan, dan 2 minggu efektif untuk menangani sekuritas sistem, atau dapat dikerjakan paralel pipeline selama 6 minggu minimal requirement. Sedangkan untuk sistem yang perfect dapat dikerjakan dengan penambahan waktu sesuai tuntutan perjanjian kerja serta problem perfect (baik in-house atau out-sourcing).

4.2. Estimasi Resources

- Resource yang dibutuhkan seperti infrastruktur jaringan telepon, komputer, modem dan segala sesuatu yang berhubungan dengan perangkat keras, tidak susah untuk didapat. Untuk sumber daya manusia dibutuhkan orang yang berpengalaman dan ahli mengenai jaringan komputer, menguasai bahasa pemrograman C++, PHP, bahasa pemrograman database seperti SQL, mengetahui salah

satu tool perancangan software seperti UML atau Power Designer, mengetahui sistem operasi Linux/UNIX dan mengetahui tentang sistem keamanan di internet (enkripsi data). Selain itu orang tersebut mampu bekerja sama dalam satu team dan mampu bekerja dalam kondisi di bawah tekanan.

- Untuk jumlah programmer yang dibutuhkan kira-kira 6 orang dengan perhitungan : 2 orang menangani database, 2 orang menangani jaringan dan 2 orang untuk security system.
- Teknologi yang tersedia sudah ada seperti bahasa pemrograman SQL untuk database, php untuk pembuatan web, alat bantu perancangan software seperti UML atau Power Designer. Sistem operasi yang mendukung server seperti linux yang stabil. Teknologi security seperti RSA 128 bit.
- Cost untuk perancangan sistem ini akan ditentukan dengan besarnya pembelian perangkat hardware, ditambah pembelian perangkat software (tools, sources) , ditambah penggajian programmer dan software

engineer selama waktu tertentu sesuai dengan besarnya kontrak kerja dengan customer, serta ditambah dengan biaya pelayanan purna jual produk (software dan hardware) berkenaan dengan garansi produk.

4.3. Risk Analysis (Software Risk)

4.3.1. Project Risk

- Resiko internal dalam mengelola proyek dapat terjadi antara lain mengembangnya resources (staffsize) dari prediksi sebelumnya karena berkembangnya masalah utama (product size).
- Resiko eksternal dapat berupa berkembangnya tuntutan customer yang merubah standard agreement, atau merubahnya pasar karena isu - isu finansial.
- Berubahnya sasaran karena dinilai mengalami kerugian kelak setelah selesai projek dalam masa pelaksanaan proyek.
- Melakukan pencatatan resiko dan penskalaan resiko secara akurat yang

mungkin terjadi selama proyek berlangsung (dalam bentuk tabel resiko) serta memberikan beberapa konsekuensi terhadap resiko proyek.

4.3.2. Technical Risk

- Kemungkinan kesalahan pemilihan metode yang ditempuh dalam menyelesaikan proyek, terhadap tinjauan sistemnya.
- Perhitungan cost yang tidak tepat sehingga menyedot dana yang mengakibatkan loss (kerugian), sehingga perlu manuver secepat mungkin tanpa mengurangi kinerja proyek/sasaran proyek yang signifikan.
- Adanya kemungkinan teknologi keamanan internet yang dipakai sudah usang dan perlu diupdate yang akan memakan waktu dan biaya.

4.3.3. Business Risk

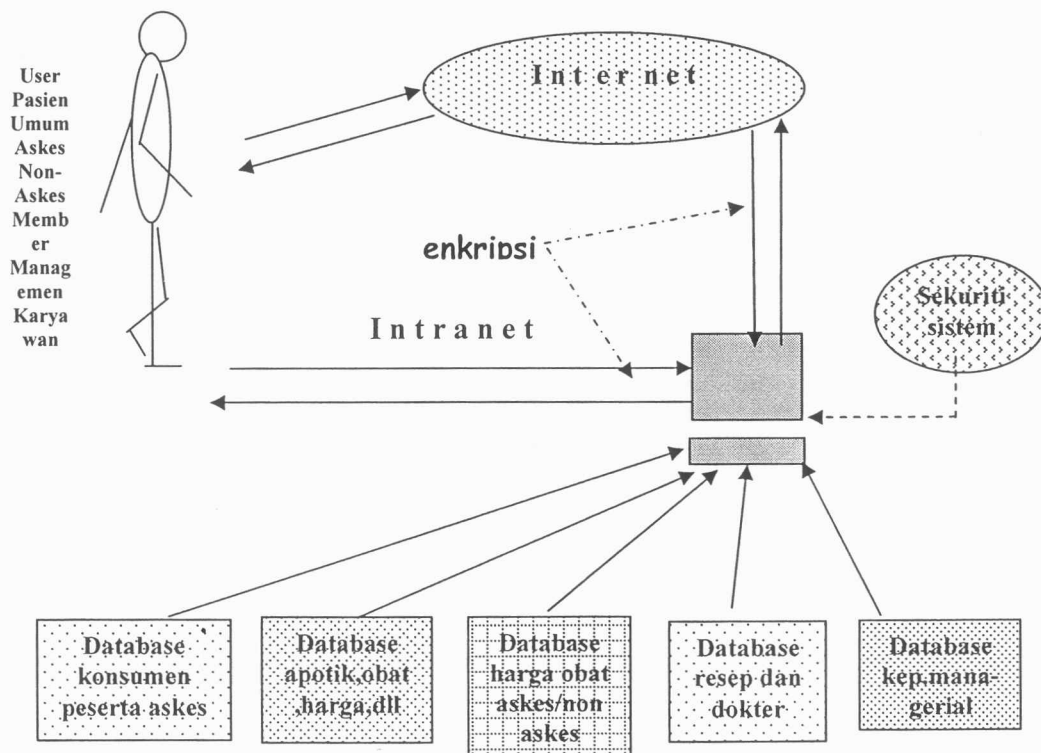
- Memprediksi secermat mungkin terhadap pelaksanaan proyek baik dari unsur usaha jasa atau problem solving, sehingga hanya terjadi resiko yang seminimal mungkin, jika terpaksa harus menelan pil pahit.
- Prediksi terhadap beberapa alternatif solving proyek yang telah diprioritaskan sehingga cepat melakukan manuver sesuai kekritisannya masalah yang telah dihadapi, jika terjadi kebuntuan.
- Menerapkan proaktif risk dengan jalan melakukan strategi management misalnya dengan identifikasi resiko dan probabilitasnya, memilih dampak yang terjadi, development environment, dan estimasi pengaruh & resiko dari proyek dan produk.

4.4. Project Scheduling

- Tabel penjadualan proyek :

| No | Action | Weekly Process | | | | | | | | | | |
|----|---|----------------|---|---|---|--------|---|---|-----------|---|----|----|
| | | First Term | | | | Medium | | | Post Term | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | Prepairing sistem (Brainstorming + Work planning) | █ | █ | | | | | | | | | |
| 2 | User requirement & User needed (SLA) | ▨ | ▨ | ▨ | | | | | | | | |
| 3 | Decomposition of Problems and Processes | | | | | | | | | | | |
| | • Data Base (SQL) | ▩ | ▩ | ▩ | ▩ | | | | | | | |
| | • Web Base (php) | ▩ | ▩ | ▩ | ▩ | ▩ | | | | | | |
| | • Encription / Security System (RSA 128 bit) | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | | | |
| | • Integration and Composition System | ▩ | ▩ | ▩ | ▩ | ▩ | ▩ | ▩ | | | | |
| | • Pengujian and Analysis System Total | ▩ | ▩ | ▩ | ▩ | ▩ | ▩ | ▩ | ▩ | ▩ | | |
| 4 | Finishing and Final step | | | | | | | | | | ▩ | ▩ |

4.5. Diagram Interkoneksi Sistem



5. Desain Perangkat Lunak

Desain perangkat lunak menggunakan pendekatan orientasi obyek dengan menggunakan Rational Rose® 2000. Data hasil pemrogramannya terlampir pada halaman terakhir laporan ini dalam bentuk hardcopy dan disertakan hasil rancangan tersebut dalam bentuk softcopy yang disertakan dengan disket.

6. Pengujian Desain

Ada tiga isu dalam hal pengujian terhadap program disain ini, yaitu lingkup pengujian (apa yang akan diuji), metode/prosedur pengujian, dan analisa hasil pengujian. Hal tersebut akan diuraikan berikut ini :

6.1. Lingkup/scoop Pengujian

Hal-hal yang menjadi fokus pengujian adalah pengujian terhadap sub-sub program, dengan cara diberikan masukan parameter dan menunjukkan hasil proses yang benar maka tak terjadi permasalahan,

kemudian dilanjutkan dengan pengujian penggabungan sub-sub program, setelah dirunning tak terjadi masalah maka dilanjutkan dengan pengujian terhadap struktur program total setelah semua sub-program digabung. Pengujian terakhir dilakukan secara bersama-sama dengan pengguna (user) sehingga dapat dicek terhadap perjanjian (user requirement), maka dengan cara ini diharapkan dapat memenuhi kriteria 'customer satisfaction' karena memenuhi permintaan user.

6.2. Metode/prosedur

Pengujian

Prosedur standard yang digunakan untuk pengujian tersebut adalah meliputi hal-hal berikut.

1. Unit Test
2. Integration Test
3. Validation Test
4. System Test

Tujuan dari prosedur ini adalah untuk menemukan sebanyak-

banyaknya kemungkinan kesalahan pada perangkat lunak tersebut dan memperbaikinya dalam waktu yang cukup realistis.

Unit Testing lebih ditekankan pada pemeriksaan sub-program (elemen terkecil dari suatu program), sedangkan Integration Testing lebih ditekankan pada pemeriksaan struktur program jika sub-program - sub-program tersebut digabung (merger) , dan Validation Test lebih ditekankan pada pemeriksaan perangkat lunak hasil desain tersebut dilihat dari sudut pandang customer/user. Serta System Test merupakan post test dari desain setelah dilakukan full composition atau integrasi total.

6.3. Analisa Hasil Pengujian

Dengan metode pendekatan orientasi obyek ini membuat mudah dan praktis bila hanya untuk melakukan pre-design suatu program. Hal ini dikarenakan adanya editor yang

cukup interaktif dan komunikatif dari mulai menentukan aktor (pelaku-pelaku dalam interaktif dengan program tersebut). Hanya ada empat tahap saja dalam pre-design perangkat lunak yaitu mulai membuat aktor dari newscase, kemudian step membuat squencer yang menyatakan alur jalannya program dan keterkaitannya dengan setiap akornya dan terhadap sistem, lantas collaboration yang merupakan interaksi antar aktor, sub-program, dalam sistem, dan diteruskan dengan class diagram yang menunjukkan tingkatan kelas dari tiap program dan sub.program dari suatu sistem yang lebih besar yang nantinya akan dikembangkan. Sehingga hasil pengujian yang perlu dianalisa adalah interaksi antar aktor, sub.program yang mungkin tak logis dan tak mungkin terjadi, kemudian penentuan aktor, parameter, besaran yang tak sesuai dengan semesta

pembicaraan, yang justru membuat ruwetnya sistem. Semua hal tersebut di atas dapat diperbaiki langsung lewat editor yang tersedia, cukup sederhana.

7. Analisis Metode yang dipakai terhadap Metode lain (Metode perancangan software yang digunakan)

7.1. Metode pendekatan Orientasi Obyek (Ratinal Rose 2000®)

Dengan metode ini menurut kami cenderung lebih praktis, karena akan mempermudah pemrogram melakukan pekerjaannya. Hal ini dikarenakan program cukup interaktif dan komunikatif serta sederhana. Tidak perlu mendiskripsikan suatu besaran dengan teliti hingga floating point-nya. Mudah cara mengedit jika terjadi suatu kesalahan, pada sub. program, karena belum terjadi integrasi total sistem antar sub. tersebut, sehingga pembenahan

kesalahan bisa dilakukan dengan terpisah antar sub.

7.2. Metode pendekatan Struktural (Power Designer ®)

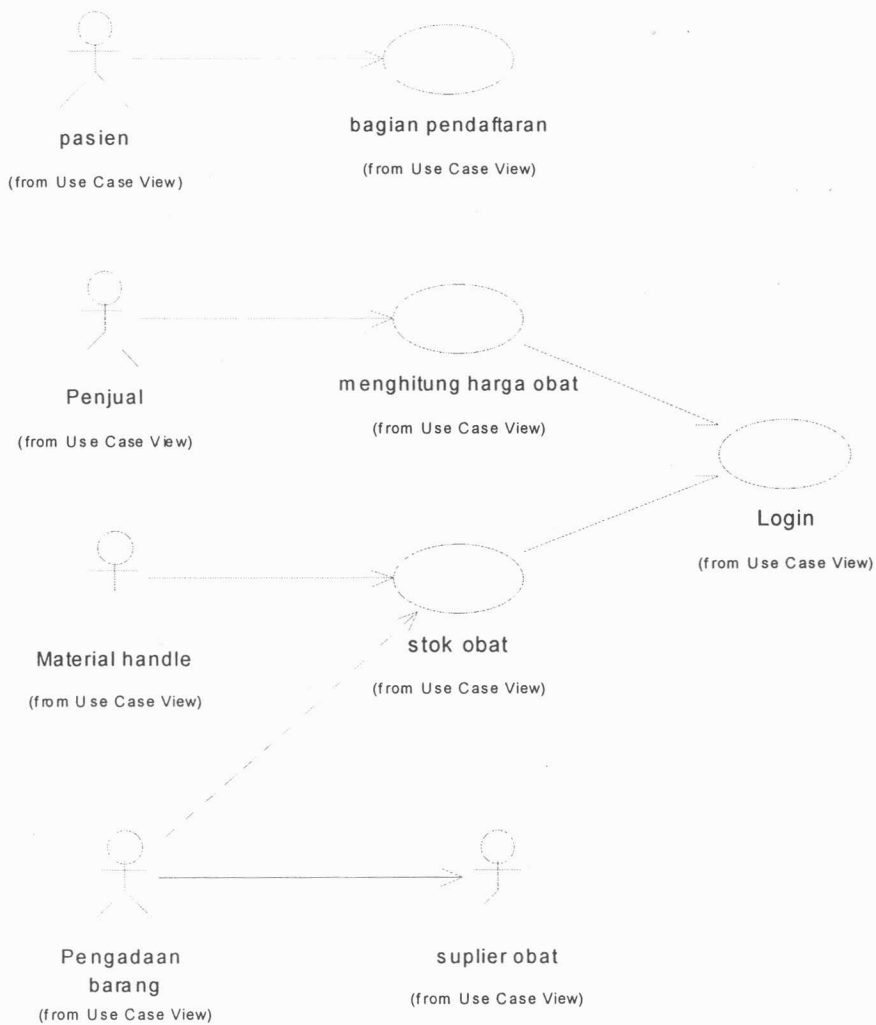
Pada metode ini cukup rumit pada satu sisinya yaitu masalah penyertaan floating desimal setiap suatu nilai besaran atau parameter, dan secara struktural merupakan hal yang wajib dipenuhi agar rancangan valid. Selain itu jika terjadi kesalahan susah melakukan editing karena setiap sub. program sudah terjadi integrasi total antara sub.program yang lain. Sehingga perlu pembenahan yang lebih teliti dan cermat agar pemrogram tak kebingungan, hal itu karena perubahan suatu bagian saja akan menyebabkan perubahan pada bagian yang lain secara berkaitan (karena terintegrasi tadi). Namun disisi lain bagi pemrogram yang telah mahir akan merasa enak melakukan pemrograman dengan pendekatan ini disebabkan lebih tegas dalam

menentukan suatu besaran atau

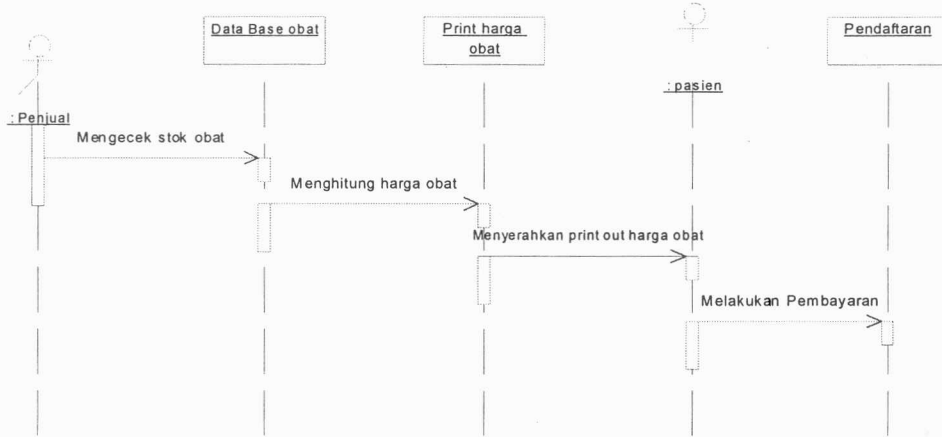
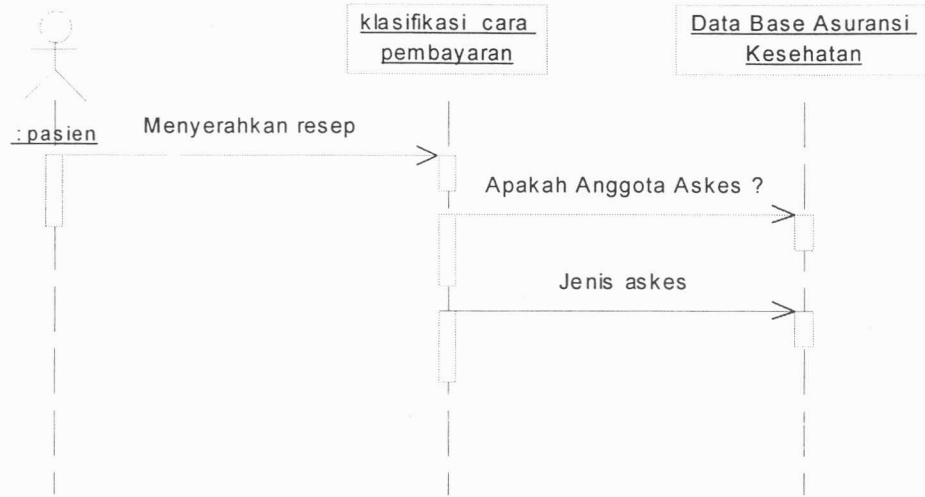
parameter sejak dini.

8. Lampiran perancangan dengan pendekatan orientasi obyek menggunakan Rational Rose 2000®

8.1. Use Case



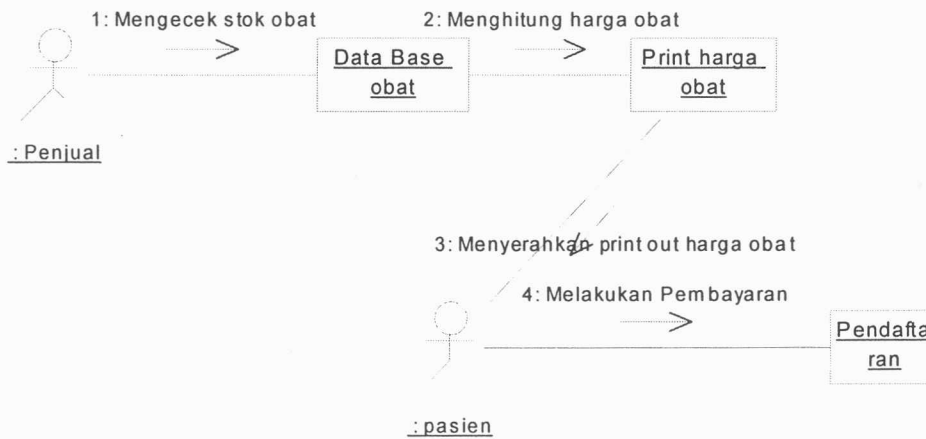
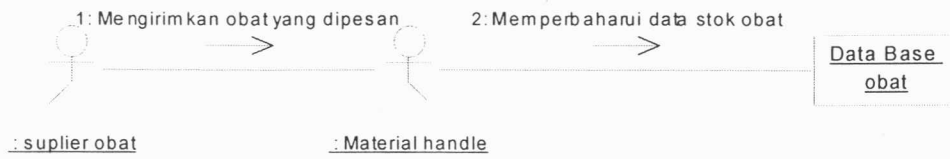
8.2. Squencer



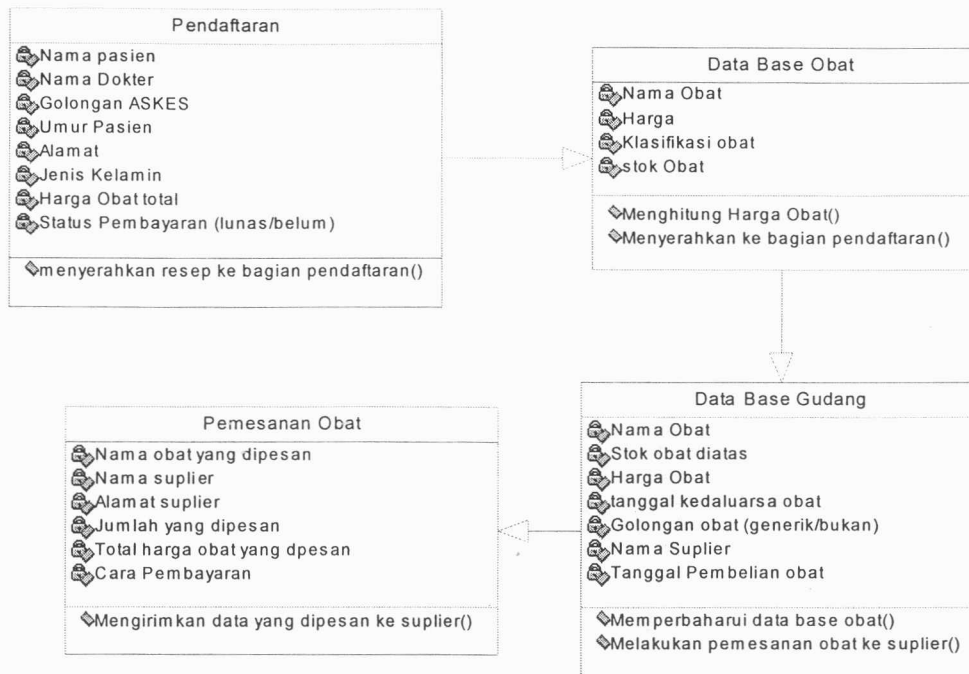


8.3. Collaboration



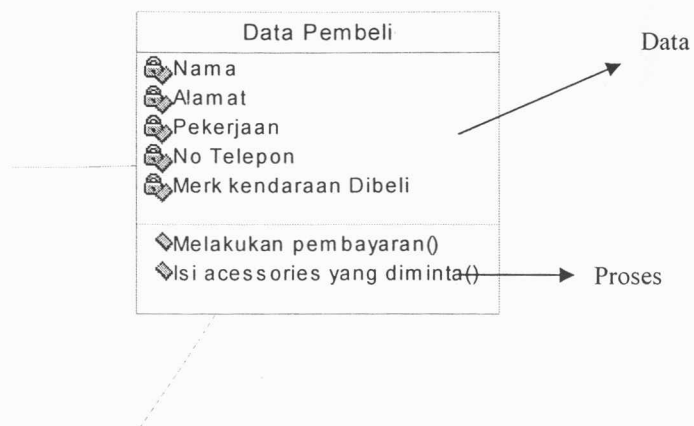


8.4. Class Diagram



9. Kesimpulan metoda Object oriented programming dibanding dengan structural programming

Seperti tampak dari model, metoda Object Oriented Programming dalam sebuah class antara data dan proses digabung. Sehingga bila kita melakukan upgrading pada software tidak akan mengganggu struktur program karena proses telah dipisahkan untuk setiap classnya. Metoda konvensional Struktur programming antara data dan proses tidak dapat digabung, sehingga bila melakukan upgrade terhadap program harus dilakukan dengan hati-hati dan pengecekan struktur program harus dilakukan secara keseluruhan karena adanya variable global.



Gambar penggabungan antara data dan proses di model OO

10. Daftar pustaka

- Roger S. Pressman, Software Engineering a practitioner's approach, Mc Graw Hill, fifth edition
- Muslihuiddin, Tutorial Singkat Rational Rose 2000, 2002
- Sapri Sutisna, pengenalan UML, Technical paper