

Aplikasi Berorientasi Objek untuk Perhitungan Bagian Waris Berdasarkan Hukum Islam

Ade Jamal, Arif Supriyanto, Fatimah Ilona Asa S., Annisa Utami, Bagus Mauludin

Program Studi Teknik Informatika, Fakultas Sains & Teknologi
Universitas Al Azhar Indonesia, Jl. Sisingamangaraja, Jakarta 12110

Penulis untuk korespondensi/Email: arifsupriyanto@uai.ac.id , adja@uai.ac.id

Abstrak – Pembagian harta waris dalam keluarga merupakan hal yang krusial yang dapat memicu pertikaian yang menimbulkan keretakan hubungan dalam keluarga. Penyebabnya adalah kurangnya pemahaman ahli waris/ pihak-pihak yang terkait mengenai ilmu faraidh (Ilmu Waris). Teknologi informasi yang berkembang pesat idealnya mampu memberikan kontribusi kepada masyarakat Muslim untuk memudahkan dalam menghitung pembagian waris. Dalam penelitian ini, dibuat Perancangan Software aplikasi berorientasi objek untuk perhitungan bagian waris berdasarkan hukum waris Islam. Di dalam pemrograman berorientasi objek terdapat beberapa fitur berikut: Enkapsulasi (Encapsulation); Abstraksi (Abstraction); Pewarisan (Inheritance); dan Polimorfisme (Polymorphism). Tampilan aplikasi terdiri dari Menu Utama yang berisi seluruh ahli waris digunakan untuk memasukkan kelompok ahli waris, pilihan hukum waris berdasarkan Hukum kompilasi Islam RI dan Mazhab Syaf'i dan Hasil Hitung berupa tampil tabel dan cetak pdf.

Abstract – The division an inheritance in the family is crucial, and has a potential to trigger conflicts in the family. The cause is a lack of understanding of the heir / related parties regarding the science of faraidh (the science of inheritance). Rapid development of information technology, ideally should be able to contribute to the Muslim community in calculating the division of inheritance. In this study, we designed an object-oriented application software to calculate inheritance based on the Islamic law. In the object-oriented programming are the following features: Encapsulation (Encapsulation); Abstraction, inheritance and polymorphism. Display application consists of a Main Menu that contains all the heirs, the choice of inheritance law based on compilation of Islamic Law in the Republic of Indonesia, and the mazhab (school) of Syaf'i result of the calculation which appear in the form of tables and print out pdf.

Keyword – inheritance law, software, inheritance, object-oriented programming, count.

PENDAHULUAN

Agama Islam adalah agama yang sempurna yang mengatur segala sisi kehidupan manusia, bahkan dalam hal yang berkaitan dengan peralihan harta yang ditinggalkan seorang manusia, setelah manusia tersebut meninggal dunia. Hal tersebut dibahas dalam suatu aturan atau hukum yang disebut hukum waris islam.

Kenyataannya pembagian harta waris dalam keluarga merupakan hal yang krusial yang dapat memicu pertikaian yang menimbulkan keretakan hubungan dalam keluarga. Penyebabnya adalah kurangnya pemahaman ahli waris/ pihak-pihak

yang terkait mengenai ilmu *faraidh*. Teknologi informasi yang berkembang pesat idealnya mampu memberikan kontribusi kepada masyarakat Muslim untuk memudahkan dalam menghitung pembagian waris. Dalam penelitian ini, kami membuat “Perancangan *Software* Aplikasi Berorientasi Objek untuk Perhitungan Bagian Waris Berdasarkan Hukum Waris Islam”.

Tujuan

Tujuan yang akan dicapai melalui penelitian ini adalah sebagai berikut :

1. Membuat *software* untuk pembagian hak waris berdasarkan hukum waris Islam.

2. Memudahkan dalam perhitungan bagian hak waris.
3. Membuat *software* yang berbasis objek sehingga lebih mudah dikembangkan.

KERANGKA TEORI

Pemrograman Berorientasi Objek^[6]

Kelas dan Objek

Sebuah kelas menentukan struktur dan *behaviour* dari sebuah objek. Objek merupakan dasar dari pemrograman berorientasi objek, di dalam dunia nyata, setiap objek yang ada memiliki dua buah karakteristik: *state* dan *behaviour*.

1. *State* merupakan atribut yang dimiliki oleh sebuah objek.
2. *Behaviour* adalah fungsi yang dimiliki dan bisa dijalankan oleh penggunanya.

Kelas merupakan sebuah cetak biru dari objek. Seorang *programmer* akan menulis kode-kode untuk tiap kelas yang akan digunakan. Sedangkan objek merupakan bentukan (*instance*) dari sebuah kelas yang dijalankan pada saat *runtime*. Pada saat *runtime*, variabel yang ada pada sebuah kelas akan menjadi *State* pada sebuah objek, sedangkan *method* yang dibuat akan menjadi *behaviour*-nya.

Terdapat beberapa keuntungan menggunakan pemrograman berorientasi objek, antara lain:

1. Real World Programming

Dunia ini disusun atas objek-objek yang saling berinteraksi. Sebuah program yang memodelkan dunia nyata, sebisa mungkin menggambarkan kondisi yang ada dalam bentuk yang seakurat mungkin. Dalam Pemrograman berorientasi objek, sebuah program disusun oleh objek-objek yang masing-masing memiliki fungsi sesuai dengan peran dan kebutuhan interaksinya.

2. Reusability of Code

Kelas yang telah dibuat dalam pemrograman berorientasi objek bisa digunakan oleh program lain. Penggunaan komponen yang telah dibuat tidak hanya mengurangi usaha pembuatan komponen tersebut, tetapi juga mengurangi kemungkinan kesalahan jika harus mengembangkan lagi dari awal. Keuntungannya adalah penghematan dari segi waktu, dan usaha yang akhirnya akan membawa kepada penghematan biaya pengembangan.

3. Resilience to Change

Dunia nyata adalah sesuatu yang dinamis, perubahan akan selalu terjadi didalamnya. Program atau aplikasi yang memodelkan dunia nyata, diharapkan juga bisa bersifat dinamis pula.

4. Information Hiding

Informasi dalam sebuah objek sedapat mungkin disembunyikan dari luar kelas yang ada. Tujuannya adalah mengamankan data agar hanya fungsi yang ada dalam kelas itu saja yang bisa membaca, mengubah dan memanipulasi data tersebut. Namun, tetap disediakan sebuah cara (*interface*) agar objek dari luar bisa mengakses dan mengubah data secara tidak langsung.

Konsep penyembunyian informasi ini bertujuan agar pihak luar yang membutuhkan layanan dari kelas tersebut hanya perlu untuk menerima data yang dibutuhkan saja, tanpa perlu mengetahui bagaimana sebenarnya cara kerja di dalam kelas tersebut.

Sebagai contoh pada sebuah jam. Kita sebagai pengguna tidak perlu memahami dengan pasti bagaimana cara kerja di dalam jam tersebut. Para pengguna tidak perlu mengetahui sebanyak apa roda giginya, menggunakan tenaga baterai atau kinetik, dan sebagainya. Yang perlu diketahui oleh pengguna adalah bahwa jam tangan tersebut menyediakan sebuah penunjuk (yang bisa kita anggap sebagai fungsi) yang menunjukkan bahwa saat ini waktu menunjukkan pukul berapa.

5. Modularity of Code

Salah satu keuntungan dari pemrograman berorientasi objek adalah modularitas, yang berarti bahwa setiap objek yang dibentuk dikelola secara terpisah dari objek lainnya meskipun berasal dari sebuah kelas yang sama. Modifikasi terhadap sebuah objek bisa dilakukan tanpa mempengaruhi fungsionalitas dari objek yang lainnya.

Kelas dan Objek^[1]

Sebuah kelas menentukan struktur dan *behaviour* dari sebuah objek. Sebagai sebuah contoh, truk, bus, mobil, sepeda motor didefinisikan sebagai sebuah kendaraan bermotor karena memiliki beberapa karakteristik yang sama yaitu memiliki mesin, menggunakan bahan bakar minyak untuk pengoperasiannya. Kendaraan bermotor tersebut juga memiliki beberapa atribut yang sama dalam kelasnya, misalkan memiliki jumlah roda, no kendaraan, nomor rangka mesin, jumlah kursi penumpang, dan lain-lain.

Fitur Pemrograman Berorientasi Objek^[6]

Didalam pemrograman berorientasi objek terdapat beberapa fitur berikut:

1. Enkapsulasi (*Encapsulation*)
2. Abstraksi (*Abstraction*)
3. Pewarisan (*Inheritance*)
4. Polimorfisme (*Polymorphism*)

1. Enkapsulasi

Enkapsulasi adalah suatu cara untuk menyembunyikan detail implementasi detail dari sebuah kelas. Terdapat dua hal mendasar dari enkapsulasi yaitu:

- a. Information hiding: penyembunyian detil dari atribut dan method pada sebuah kelas.
- b. Interface untuk pengaksesan data: suatu method untuk mengambil, memberikan atau mengubah suatu nilai.

Dalam pemrograman berorientasi objek, kebutuhan akan enkapsulasi muncul karena adanya proses sharing data antar method. Dengan menggunakan enkapsulasi, maka keamanan dari data dan method yang ada didalamnya akan terjaga.

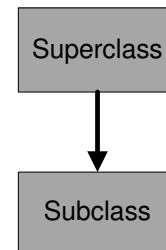
2. Abstraksi

Arti abstraksi mengacu kepada atribut dari sebuah objek yang membedakan antara satu objek dengan objek yang lain. Misalnya pada sebuah sistem akademik, dimana terdapat beberapa komponen misal saja mahasiswa. Didalam objek tersebut, terdapat suatu atribut yang akan saling membedakan antara satu objek mahasiswa dengan mahasiswa lainnya, yaitu NIM, Nama, tanggal lahir, alamat, dan sebagainya. Dalam pemrograman berorientasi objek konsep ini berada pada pembuatan sebuah kelas. Semua atribut dari objek didefinisikan dalam sebuah kelas. Sebenarnya kelas tidak memiliki data, tetapi sebuah objek-lah yang akan menyimpan data, karena objek diciptakan dari sebuah kelas dan oleh sistem operasi akan dialokasikan sejumlah memori kepada objek tersebut.

3. Pewarisan

Salah satu fitur yang paling kuat dalam pemrograman berorientasi objek adalah penggunaan kode kembali (*code reuse*). Sekali sebuah prosedur dibuat, maka kita bisa menggunakannya berulang kali. Dalam pemrograman berorientasi objek, kemampuan yang dimiliki tidak hanya itu, tetapi kita juga bisa mendefinisikan hubungan antar kelas yang tidak hanya dimanfaatkan untuk *code reuse*, tetapi juga dari segi pendesainannya yang secara garis besar

jauh lebih baik, dengan cara mengelola kelas-kelas dan faktor kemiripan diantara kelas-kelas tersebut. Tujuan utama dari pewarisan (*inheritance*) adalah untuk menyediakan fungsionalitas tersebut.

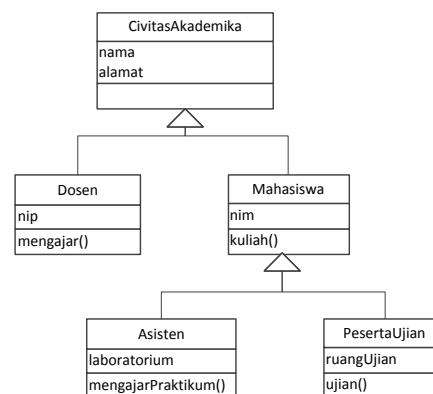


Gambar 1 : *Superclass* dan *Subclass*

Fitur pewarisan memungkinkan sebuah kelas yang dinamakan *superclass* untuk menurunkan atribut-atribut dan methodnya kepada yang lainnya, yaitu yang disebut *subclass* atau kelas turunannya. Hal ini akan memungkinkan pembuatan kelas baru yang didasarkan dari peng-abstrakan atribut-atribut dan *behaviour* yang sama.

Terdapat beberapa karakteristik dari *subclass*, yaitu:

1. *Subclass* merupakan bentuk khusus dari sebuah *superclass*.
2. Dalam sebuah *subclass* dapat memiliki atribut dan *method* yang diturunkan dari sebuah *superclass*.
3. *Subclass* bisa memiliki fitur tambahan dan berbeda yang berbeda dari fitur-fitur yang diturunkan dari *superclass*-nya.



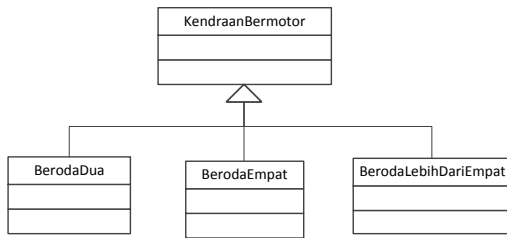
Gambar 2 : Contoh pewarisan

Dari contoh diatas bisa dilihat bahwa terdapat sebuah kelas Civitas Akademik yang memiliki 2 (dua) kelas turunan, yaitu Dosen dan Mahasiswa. Meskipun didalam kelas Dosen dan Mahasiswa tidak tercantum atribut Nama dan Alamat, tetapi keduanya tetap memiliki kedua atribut tersebut yang diturunkan dari kelas CivitasAkademik.

Begitu juga untuk kelas Asisten dan PesertaUjian. Keduanya tetap memiliki atribut Nama dan Alamat yang diturunkan dari kelas Mahasiswa, yang sebenarnya juga turunan dari kelas Civitas Akademik. Hal tersebut tidak hanya berlaku untuk atribut saja, tetapi juga *method-method* yang ada di kelas induknya.

Didalam pemrograman berorientasi objek, terdapat beberapa jenis pewarisan, yaitu:

1. *Single inheritance*, yaitu hanya terdapat satu *superclass*.



Gambar 3 : *Single inheritance*

2. *Polimorfisme*

Polimorfisme diturunkan dari bahasa latin yaitu poly yang berarti banyak dan morph yang berarti bentuk. *Polimorfisme* sendiri berarti sesuatu yang memiliki banyak bentuk. Sebagai contoh adalah sebuah objek wanita, beberapa peran yang mungkin dimiliki adalah:

- a. Bagi suami maka dia berperan sebagai seorang istri.
- b. Buat anak-anak berperan sebagai ibu.
- c. Di tempat kerja maka dia akan berperan sebagai seorang karyawan.
- d. Di tempat kuliah berperan sebagai mahasiswi.
- e. Di tempat arisan berperan sebagai ketua arisan.

Dari contoh diatas bisa kita lihat bahwa wanita tersebut adalah orang yang sama, tetapi memiliki peran yang berbeda bagi orang yang berinteraksi dengannya.

Didalam pemrograman berorientasi objek, *polimorfisme* adalah sebuah fitur yang memungkinkan kita untuk memberikan arti atau penggunaan yang berbeda bagi sebuah entitas dalam konteks yang berbeda-beda. Entitas tersebut bisa berupa variabel, *method*, atau sebuah objek. *Polimorfisme* bisa digunakan sebagai kategori umum bagi sebuah entitas dalam tindakan yang berbeda-beda.

Unified Modeling Language (UML)^[2]

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan dan desain program berorientasi objek (OOP) serta aplikasinya.

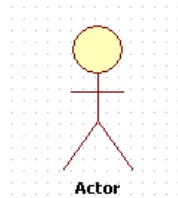
1. *Use Case Diagram*

Use case diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang menekankan pada apa yang dapat diperbuat oleh sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem, misalnya proses pendaftaran *member* baru, proses peminjaman buku, dan sebagainya.

Use case diagram adalah gambaran *graphical* dari beberapa atau semua aktor, *use case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar. *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. Berikut adalah komponen-komponen yang menjadi pembentuk *Use case diagram* :

a. *Actor*

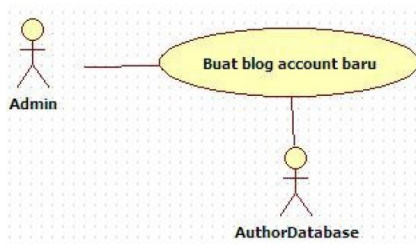
Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.



Gambar 4 : Actor

b. Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.



Gambar 5 : Contoh Use case diagram

2. Class Diagram

Class diagram adalah diagram UML yang menggambarkan struktur dari sebuah sistem yang dibuat dari kelas-kelas dan relasinya. kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi.

Class diagram mendeskripsikan jenis jenis objek dalam sistem dan berbagai macam hubungan statis yang terjadi. class diagram juga menunjukkan properti dan operasi sebuah class dan batasan yang terdapat dalam hubungan dengan objek. Class diagram merupakan alat terbaik dalam perancangan perangkat lunak. Class Diagram membantu pengembang mendapatkan struktur sistem dan menghasilkan rancangan sistem yang baik. Class memiliki tiga area pokok :

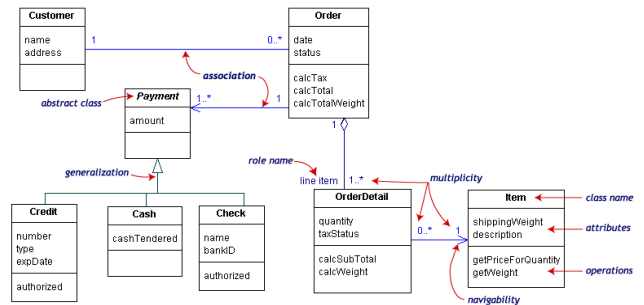
- a. Nama
- b. Attribute
- c. Metode

Attribute dan metode dapat memiliki salah satu sifat berikut :

- a. Private, tidak dapat dipanggil dari luar class yang bersangkutan.
- b. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
- c. Public, dapat dipanggil oleh siapa saja.

Hubungan antar Class:

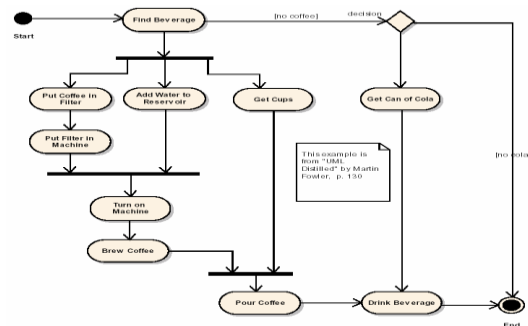
- a. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki attribute berupa class lain atau class yang harus mengetahui eksistensi class lain.
- b. Agregasi, yaitu hubungan yang menyatakan bagian dari
- c. Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari Class yang diwarisinya.



Gambar 6 : Class diagram

3. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, hasil akhir yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.



Gambar 7 : Activity diagram

4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk

menghasilkan output tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakan antara objek-objek ini di dalam *use case*.



Gambar 8. contoh *Sequence Diagram*

System Development Life Cycle^[5]

1. Perencanaan sistem
 - a. Dibentuk suatu struktur kerja strategis yang luas dan pandangan sistem informasi baru yang jelas yang akan memenuhi kebutuhan-kebutuhan pemakai informasi.
 - b. Proyek sistem dievaluasi dan dipisahkan berdasarkan prioritasnya. Proyek dengan prioritas tertinggi akan dipilih untuk pengembangan.
 - c. Sumber daya baru direncanakan untuk, dan dana disediakan untuk mendukung pengembangan system.
2. Analisis sistem
 - a. Dilakukan proses penilaian, identifikasi dan evaluasi komponen dan hubungan timbal-balik yang terkait dalam pengembangan sistem; definisi masalah, tujuan, kebutuhan, prioritas dan kendala-kendala sistem; ditambah identifikasi biaya, keuntungan dan estimasi jadwal untuk solusi yang berpotensi.
 - b. Perancangan sistem secara umum/konseptual.
 - c. Model struktur data potensi ahli waris adalah memodelkan struktur data yang akan

digunakan untuk memasukan calon ahli waris

- d. Desain berorientasi objek adalah mendesign model berorientasi objek sehingga mudah dikembangkan dan diperbaiki.
3. Perancangan sistem secara detail:
 - a. Fase perancangan sistem secara detail menyediakan spesifikasi untuk perancangan secara konseptual. Pada fase ini semua komponen dirancang dan dijelaskan secara detail.
 - b. Pemrograman perangkat lunak dan implementasi sistem.
 - c. Sistem siap untuk dibuat dan diinstalasi.
 4. Testing dan evaluasi sistem

MVC (*Model, View, Controller*)^[2]

MVC merupakan kependekan dari *Model View Controller*, yang merupakan sebuah metode yang didengungkan oleh peneliti XEROX PARAC yang bekerja dalam pembuatan bahasa pemrograman Smalltalk sekitar tahun 1970-1980. MVC sendiri merupakan sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*view*) dan bagaimana prosesnya (*controller*). Banyak pengembangan aplikasi dibuat tanpa memikirkan kaedah-kaedah MVC, dimana terkadang sebuah aplikasi dibuat dengan menggabung atau menyatukan data (*model*), tampilan (*view*), dan prosesnya (*controller*). Selain contoh penggabungan *Model-View-Controller* menjadi satu bagian, ada beberapa kasus pembuatan aplikasi hanya memisahkan data (*model*), dan menggabung tampilan (*view*) dengan prosesnya (*controller*) saja. Sehingga apabila sebuah aplikasi dipindah tugaskan kepada pengembang lain, maka pengembang baru yang meneruskan atau mengembangkan kembali aplikasi tersebut akan memiliki kesulitan dalam hal seperti, kesulitan untuk melacak apabila terjadi kesalahan dalam aplikasi yang dibuat, dalam arti pengembang baru yang akan mengembangkan aplikasi yang bersangkutan akan membutuhkan waktu yang lebih dari biasanya untuk memahami bagaimana aplikasi tersebut bekerja, dan menemukan kesalahannya.

Dengan mempelajari untuk menerapkan konsep MVC ini dalam pembuatan aplikasi apapun, dirasa adalah sebagai salah satu jalan terbaik yang dapat dilakukan demi kemudahan yang dihasilkan. Karena ketika berbicara mengenai desain aplikasi berorientasi objek, kita akan berbicara mengenai *loose couple* atau ketergantungan setiap *class*

terhadap *class* lainnya. Ketergantungan antar *class* sebaiknya dijaga seminimum mungkin, atau mungkin jika memungkinkan dibuat agar tidak ada ketergantungan sama sekali.

Apabila ketergantungan yang dimiliki antar *class* cukup tinggi, maka semakin erat keterkaitan antar keduanya yang akan membuahkan masalah pada proses pengembangan aplikasi pada tingkatan selanjutnya. Apabila pada proses pengembangan aplikasi pada tingkat selanjutnya terdapat perubahan-perubahan yang harus dilakukan, maka sebuah perubahan kecil pada *class* tertentu akan mengakibatkan perubahan juga pada *class* lainnya yang memiliki ketergantungan atau keterkaitan terhadap *class* yang mengalami perubahan. Ketergantungan antar dua *class* atau lebih dapat terjadi apabila *class* yang dirancang memisahkan antar pekerjaan yang sesungguhnya dapat dijadikan atau digabungkan menjadi satu. Sehingga mengakibatkan apabila terdapat perubahan pada satu *class* akan mengakibatkan perubahan pada *class* lainnya yang memiliki keterkaitan dengan *class* tersebut. Pada aplikasi kecil yang tidak memerlukan *loose coupling* pada model, apabila terjadi perubahan, maka perubahan yang dirubah hanya akan berdampak pada beberapa *class* saja. Tapi tentu akan berbeda jika aplikasi yang dirancang merupakan aplikasi dengan skala besar. Salah satu cara yang dapat dilakukan untuk meminimalisir ketergantungan antar *class* adalah dengan memilah-milah dan membebaskan tanggung jawab tiap-tiap *class* yang akan dibuat. Setiap *class* yang akan dibuat dipikirkan terlebih dahulu apa yang dapat dilakukan olehnya. Dengan pemisahan tanggung jawab berdasarkan *class*-nya, diharapkan dapat meminimalisir ketergantungan atau keterkaitan antar *class* tertentu.

Hukum Waris^[4]

Definisi Ilmu Waris Islam

Al-Mawarits (Ilmu waris islam) adalah masalah-masalah pembagian harta warisan, atau disebut juga sebagai ilmu *Al-Faraidh*. Sedangkan secara terminologis, ilmu *Al-Mawarits* memiliki beberapa definisi, diantaranya :

1. Kaidah-kaidah fiqh dan cara menghitung untuk mengetahui bagian setiap ahli waris dari harta peninggalan.
2. Ilmu yang digunakan untuk mengetahui ahli waris yang dapat mewarisi dan yang tidak dapat mewarisi serta mengetahui kadar bagian setiap ahli waris.

Dengan demikian, Ilmu *Al-Mawarits* mencakup tiga unsur penting didalamnya :

1. Pengetahuan tentang kerabat-kerabat yang menjadi ahli waris;
2. Pengetahuan tentang bagian setiap ahli waris;
3. Pengetahuan tentang cara menghitung yang dapat berhubungan dengan pembagian harta warisan.

Sumber Hukum Waris Islam^[3]

Hukum waris dalam islam adalah bagian dari syarat Islam yang bersumber dari :

1. Sumber Al-Qur'an
 - a. Q.S. An-Nisaa': ayat 7.
 - b. Q.S. An-Nisaa': ayat 11.
 - c. Q.S. An-Nisaa': ayat 12.
 - d. Q.S. An-Nisaa': ayat 13.
 - e. Q.S. An-Nisaa': ayat 14.
 - f. Q.S. An-Nisaa': ayat 176.
2. Hadits
3. Ijma' sahabat

Rukun Waris

Rukun waris terdiri dari tiga rukun. Jika salah satu dari rukun waris ini tidak ada maka tidak akan terjadi pembagian warisan. Rukun tersebut adalah :

1. Pewaris, yakni orang yang meninggal dunia dan ahli warisnya berhak untuk mewarisi harta peninggalanya.
2. Ahli waris, yaitu mereka yang berhak untuk menguasai atau menerima harta peninggalan pewaris dikarenakan adanya ikatan kekerabatan (nasab) atau ikatan pernikahan, atau lainnya.
3. Harta warisan, yaitu segala jenis benda atau kepemilikan yang ditinggalkan pewaris, baik berupa uang, tanah, dan sebagainya.

Syarat Waris

Syarat-syarat waris juga ada tiga:

1. Meninggalnya seseorang (pewaris) baik secara hakiki maupun secara hukum.
2. Adanya ahli waris yang masih hidup secara nyata pada waktu pewaris meninggal dunia.
3. Seluruh ahli waris telah diketahui secara pasti, termasuk kedudukannya terhadap pewaris dan jumlah bagiannya masing-masing.

Sebab Mendapat Hak Waris

Terdapat tiga sebab yang menjadikan seseorang mendapatkan hak waris, yaitu :

1. Adanya ikatan kekerabatan.
2. Adanya ikatan pernikahan.
3. Al Wala.

Penggugur Hak Waris

Penggugur hak waris seseorang maksudnya kondisi yang menyebabkan hak waris seseorang menjadi gugur, dalam hal ini ada tiga, budak, pembunuhan dan perbedaan agama.

Kelompok Ahli Waris

Ada beberapa macam kelompok ahli waris, diantaranya :

Ahli Waris Dari Golongan Laki-Laki

Ahli waris (yaitu orang yang berhak mendapatkan warisan) dari kaum laki-laki ada lima belas, yaitu anak laki-laki, cucu laki-laki (dari anak laki-laki), bapak, kakek (dari pihak bapak), saudara kandung laki-laki, saudara laki-laki seayah, saudara laki-laki dari seibu, anak laki-laki dari saudara kandung laki-laki, anak laki-laki dari saudara laki-laki seibu, paman (saudara kandung bapak), paman (saudara bapak seayah), anak laki-laki dari paman (saudara kandung ayah), anak laki-laki paman seayah, suami, dan laki-laki yang memerdekakan budak.

Bagi cucu laki-laki yang disebut sebagai ahli waris didalamnya tercakup cicit (anak dari cucu) dan seterusnya, yang penting laki-laki dan dari keturunan anak laki-laki. Begitu pula yang dimaksud dengan kakek dan seterusnya.

Ahli Waris Dari Golongan Wanita

Adapun ahli waris dari kaum wanita adalah sepuluh, yaitu anak perempuan, ibu, anak perempuan (dari keturunan anak laki-laki), nenek (ibu dari ibu), nenek (ibu dari bapak), saudara kandung perempuan, saudara perempuan seayah, saudara perempuan seibu, istri, perempuan yang memerdekakan budak.

Cucu perempuan yang dimaksud diatas mencakup pula cicit dan seterusnya, yang penting perempuan dari keturunan anak laki-laki. Demikian pula yang dimaksud dengan nenek-baik ibu dari ibu maupun ibu maupun ibu dari bapak- dan seterusnya.

Hak Ahli Waris**Ahli Waris Dengan Bagian Tertentu (*Ashhab Al-Furudh*)**

Ahli waris ini menerima bagian-bagian tertentu yang ditetapkan dalam Al-Qur'an dan Hadits, seperti bagian-bagian 1/2, 1/3, 1/4, 1/6, 1/8 dan 2/3 sebagaimana telah ditentukan dalam surat An-Nisa' ayat 11, 12, 176.

Ahli Waris Dengan Bagian Tak Tertentu (*Al-Ashabah*)

Secara bahasa berarti kerabat laki-laki dari ayah, dinamakan demikian karena mereka melingkupinya pada waktu berkumpul untuk menjaga dan melindunginya. Menurut istilah fiqh ialah setiap ahli waris yang tidak mempunyai bagian tertentu (*'Al-Furudh Al-Muqaddarah'*) pada sebuah penentuan hak waris. Atau dengan kata lain, ahliwaris yang terkadang mengambil semua harta, terkadang mengambil sisa dan terkadang tidak mendapatkan apa-apa.

Hajb

Yaitu terhalang untuk mendapatkan hak waris. Al-Hajb secara garis besar ada 2, yaitu :

1. *Al-Hajb Al-Washfiy* 'terhalang karena sifat' seperti pembunuh, hamba dan kafir atau murtad.
2. *Al-Hajb Asy-Syakhshiy* 'terhalang karena orang lain' seperti kakek dihalangi oleh ayah si mayit.

ANALISIS PERMASALAHAN

Dalam mengembangkan *software* Hukum Waris diperlukan pengetahuan dan informasi yang diperoleh beberapa sumber, diantaranya adalah orang yang ahli dan beberapa buku. Untuk mengatasi hal itu, perlu dipikirkan suatu *software* aplikasi untuk membagi harta warisan sesuai hukum islam, diharapkan dapat membantu masyarakat untuk mengetahui informasi bagian-bagian hak ahli waris sehingga akan membantu mempermudah melakukan pembagian harta warisan sesuai ajaran agama Islam.

Acuan Informasi

Acuan informasi dilakukan dengan melakukan wawancara kepada M. Rizal, S.H. seorang Hakim Pengadilan Agama, Ali Fauzan, Lc. Seorang konsultan syariah di kementerian Agama RI, dan melakukan studi literatur yang direkomendasikan beliau, yaitu buku Panduan Praktis Pembagian Waris Dalam Islam yang disusun oleh Direktorat Urusan Agama Islam dan Pembinaan Syariah Kementerian Agama Republik Indonesia dan buku Hukum Waris Terlengkap penyusun Komite Fakultas Syariah Universitas Al Azhar, Mesir.

Pembagian Harta Warisan Menurut Tabel Waris

Tabel 1. Aturan waris 1

Anak laki-laki					A
Cucu laki-laki				A	A
					0
Ayah			A	A	A
				1/4	1/4
Datuk		A	A	A	A
			0	1/6	1/6
Anak Perempuan	1/2	1/2 A	1/6	A	A
		1/2	1/2	1/2	

Contoh Pemakaian Tabel

Jika ahli waris sendiri :

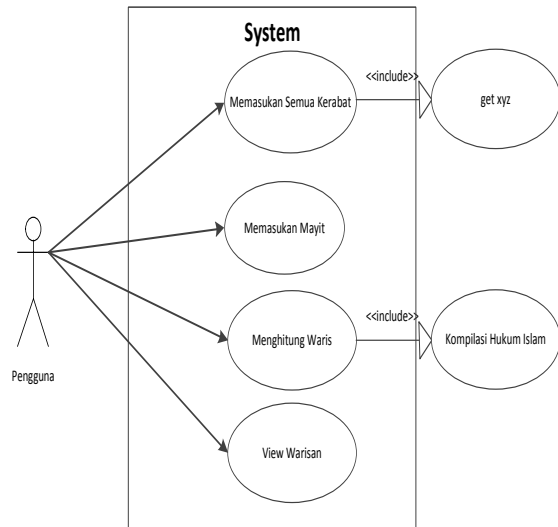
1. Pada gambar tabel Anak laki-laki hanya terdapat Nilai A maka pada tabel tersebut mempunyai keterangan bahwa Anak laki-laki sendiri atau tidak ada ahli waris lain maka mendapatkan nilai A (*Asobah*, yaitu orang yang memperoleh sisa harta warisan).
2. Pada gambar tabel Cucu laki-laki hanya terdapat Nilai A maka pada tabel tersebut mempunyai keterangan bahwa Cucu laki-laki sendiri atau tidak ada ahli waris lain maka mendapatkan nilai A (*Asobah*).
3. Pada gambar tabel Ayah hanya terdapat Nilai A maka pada tabel tersebut mempunyai keterangan bahwa Ayah sendiri atau tidak ada ahli waris lain maka mendapatkan nilai A (*Asobah*).
4. Pada gambar tabel Datuk/ Kakek hanya terdapat Nilai A maka pada tabel tersebut mempunyai keterangan bahwa Datuk/ Kakek sendiri atau tidak ada ahli waris lain maka mendapatkan nilai A (*Asobah*).
5. Pada gambar tabel Anak Perempuan hanya terdapat Nilai 1/2 maka pada tabel tersebut mempunyai keterangan bahwa Anak Perempuan sendiri atau tidak ada ahli waris lain maka mendapatkan nilai 1/2.

PERANCANGAN SISTEM

Perancangan adalah penggambaran, perencanaan, dan pembuatan sketsa untuk memberikan gambaran tentang implementasi *Software* Mawaris untuk pembagian harta waris menurut hukum Islam [7].

Use Case

Sistem yang akan dirancang melibatkan satu aktor yang bertugas menjalankan sistem tersebut. Pengguna merupakan yang menggunakan *software* tersebut. *Use case* dapat dilihat Gambar 9.

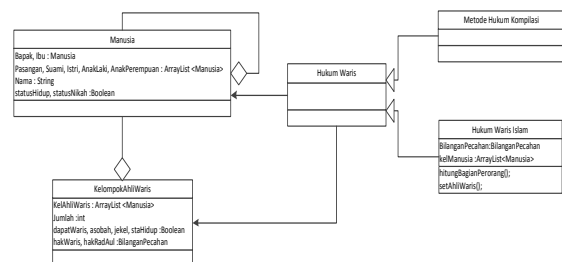


Gambar 9. Use case

Keterangan : get xyz adalah membuat objek diagram.

Kelas Diagram

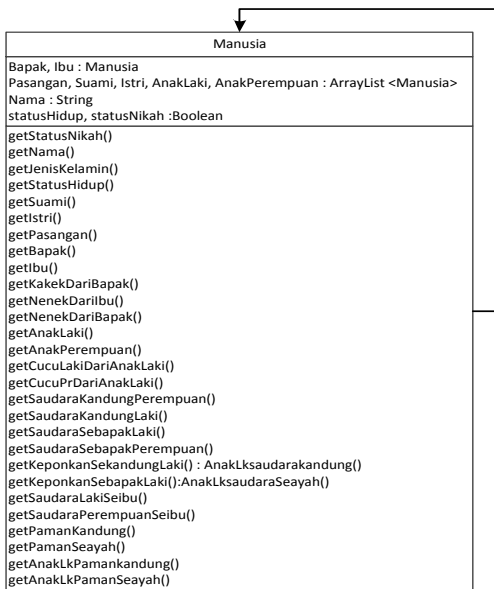
Kelas diagram berikut menjelaskan secara garis besar mengenai semua aplikasi yang terlibat.



Gambar 10. Kelas diagram

Kelas Manusia

Kelas Manusia menjelaskan tentang atribut yang terdapat pada manusia dan membentuk silsilah ahli waris dan kelompok ahli waris.



Gambar 11. Kelas manusia

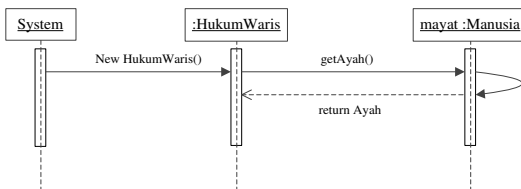
Objek Diagram

Objek digram menjelaskan tentang tersusunnya hubungan kekerabatan dapat dilihat pada Gambar 12 (lampiran).

Sequence Diagram

Sequence diagram memetakan interaksi dan message antar objek komunikasi data diantara objek-objek. Kali ini, penulis akan memetakan sequence diagram untuk pembuatan objek.

Sequence ayah menjelaskan tentang pembentukan objek ayah yaitu dari sistem membuat kelas hukum waris, mengambil ayah dari manusia yang berhubungan dengan mayat dan memberikan objek ayah.



Gambar 13. Diagram sequence ayah

Diagram Model View Control

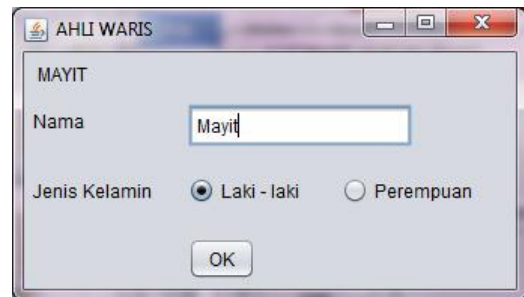
Diagram model view control merupakan digram yang menjelaskan hubungan yang terjadi antara problem domain model dengan user interface serta controller yang dapat dilihat pada Gambar 14.

Tampilan Pengguna (User Interface)

Menu Utama

Menu Utama (Gambar 14) adalah tampilan yang berisi seluruh ahli waris digunakan untuk memasukan kelompok ahli waris, pilihan hukum waris berdasarkan Hukum kompilasi Islam RI dan Mazhab Syaf'i dan Hasil Hitung berupa tampil tabel dan cetak pdf. Kelompok Ahli Waris yaitu Mayat, Suami atau Istri, Ibu, Ayah, Nenek Dari Ibu, Nenek Dari Ayah, Kakek Dari Ayah, Paman Sekandung, Paman Sebapak, Saudara Perempuan Seibu, Saudara Laki-laki Seibu, Saudara Perempuan Kandung, Saudara Laki-laki Kandung, Saudara Perempuan Sebapak, Saudara Laki-laki Sebapak, Sepupu laki-laki dari Paman Sekandung, Sepupu Laki-laki dari Paman Sebapak, Anak laki-laki, Anak Perempuan, Keponakan laki-laki dari Saudara laki-laki kandung, keponakan laki-laki dari saudara laki-laki saudara laki-laki sebapak, cucu perempuan dari anak laki-laki, cucu laki-laki dari anak laki-laki.

Menu Mayat



Gambar 15. Tampilan input mayat

Input Mayat adalah tampilan yang berisi nama mayat dan jenis kelamin, karena dengan jenis kelamin bisa menentukan ahli warisnya yaitu suaminya atau istrinya.

Uji Coba

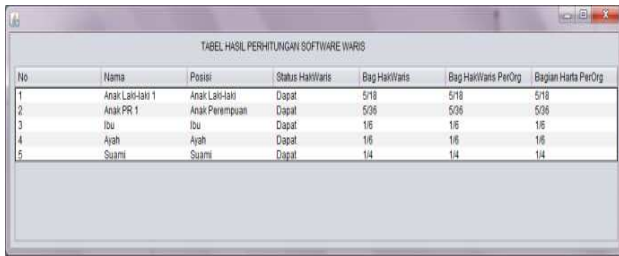
Kasus 1

Jika seorang istri meninggal dunia, meninggalkan Ayah, Ibu, Suami, 1 Anak Laki- laki, satu Anak Perempuan Maka :

Penyelesaian Dari Buku

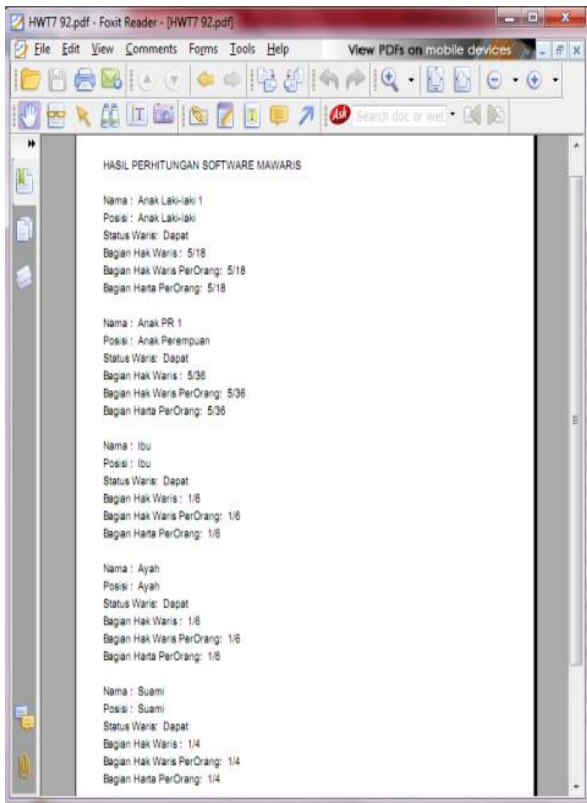
Maka :
 Ayah = 1/6
 Ibu = 1/6
 Suami = 1/4
 Anak laki-laki = Sisa = 10/36= 5/18
 Anak Perempuan = Sisa = 5/36

Hasil Menggunakan Software Hukum Waris



No	Nama	Posisi	Status HakWaris	Bag HakWaris	Bag HakWaris PerOrg	Bagian Harta PerOrg
1	Anak Laki-laki 1	Anak Laki-laki	Dapat	5/18	5/18	5/18
2	Anak PR 1	Anak Perempuan	Dapat	5/36	5/36	5/36
3	Ibu	Ibu	Dapat	1/6	1/6	1/6
4	Ayah	Ayah	Dapat	1/6	1/6	1/6
5	Suami	Suami	Dapat	1/4	1/4	1/4

Gambar 16. Tampilan hasil perhitungan *software* hukum waris



HASIL PERHITUNGAN SOFTWARE MAWARIS

Nama : Anak Laki-laki 1
Posisi : Anak Laki-laki
Status Waris : Dapat
Bagian Hak Waris : 5/18
Bagian Hak Waris PerOrang : 5/18
Bagian Harta PerOrang : 5/18

Nama : Anak PR 1
Posisi : Anak Perempuan
Status Waris : Dapat
Bagian Hak Waris : 5/36
Bagian Hak Waris PerOrang : 5/36
Bagian Harta PerOrang : 5/36

Nama : Ibu
Posisi : Ibu
Status Waris : Dapat
Bagian Hak Waris : 1/6
Bagian Hak Waris PerOrang : 1/6
Bagian Harta PerOrang : 1/6

Nama : Ayah
Posisi : Ayah
Status Waris : Dapat
Bagian Hak Waris : 1/6
Bagian Hak Waris PerOrang : 1/6
Bagian Harta PerOrang : 1/6

Nama : Suami
Posisi : Suami
Status Waris : Dapat
Bagian Hak Waris : 1/4
Bagian Hak Waris PerOrang : 1/4
Bagian Harta PerOrang : 1/4

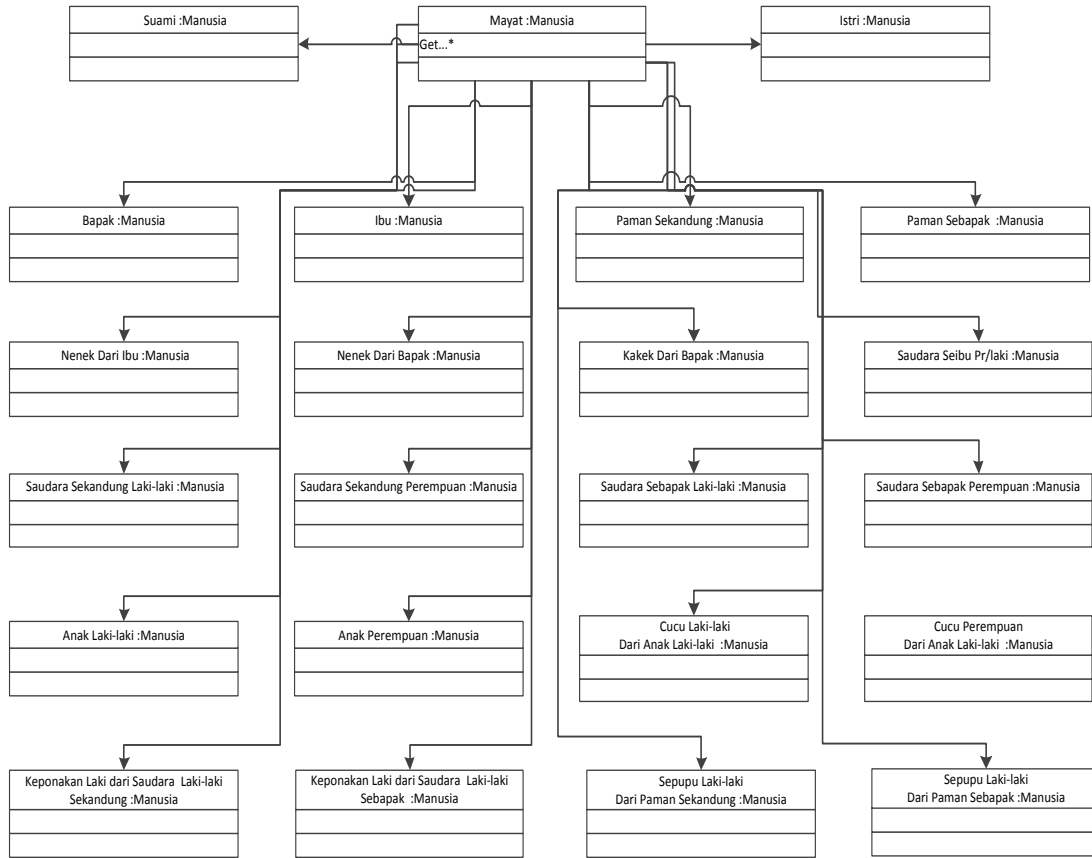
Gambar 17. Tampilan print PDF hasil perhitungan *software* hukum waris

KESIMPULAN DAN SARAN

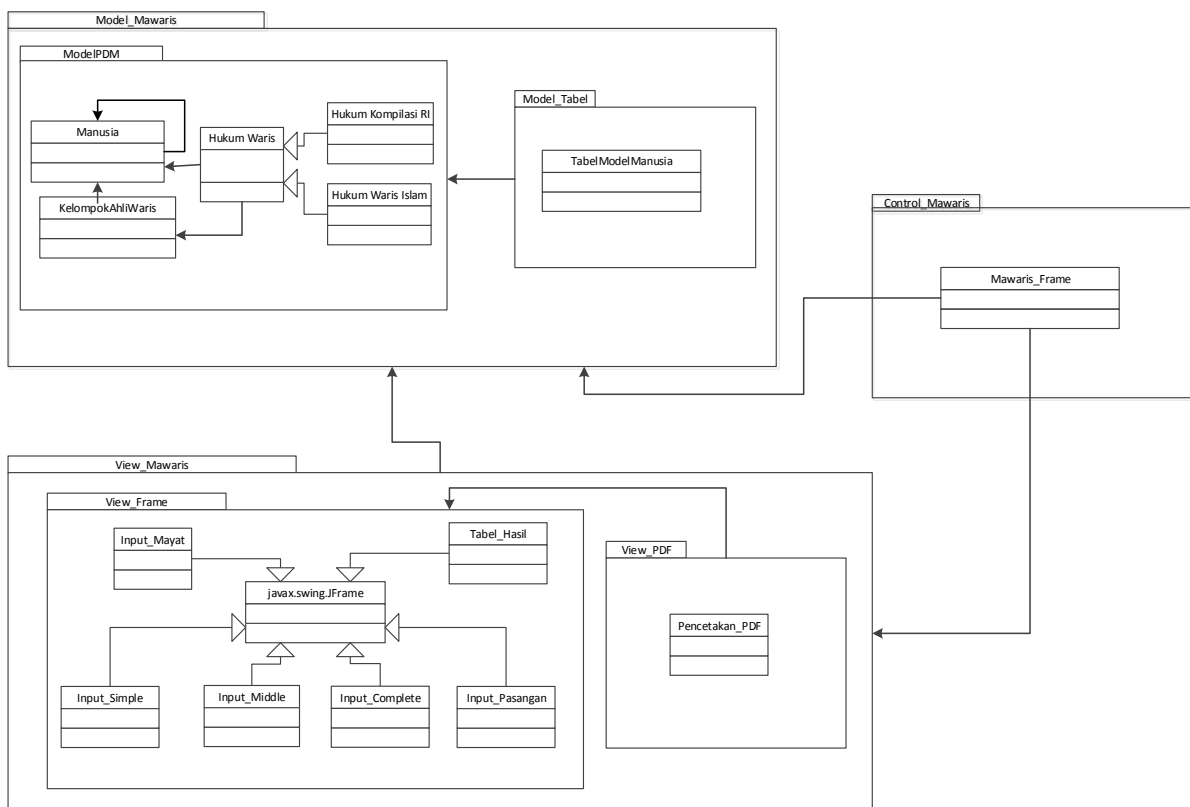
1. *Software* hukum waris berhasil melakukan perhitungan bagian hak ahli waris dengan bilangan pecahan sehingga lebih akurat.
2. *Self aggregation* mempermudah untuk membuat silsilah keluarga pada hubungan kekeluargaan ahli waris.
3. Penerapan pilihan hukum waris yang berbeda bisa diimplementasikan dengan diturunkan dari kelas hukum waris sehingga memudahkan pengguna untuk menentukan hak waris sesuai hukum yang dipilih.
4. Pilihan hukum waris dikembangkan dengan berorientasi objek sehingga mudah dikembangkan untuk hukum waris lain.
5. Aplikasi ini dapat dikembangkan dengan mengembangkannya menjadi berbasis web.

DAFTAR PUSTAKA

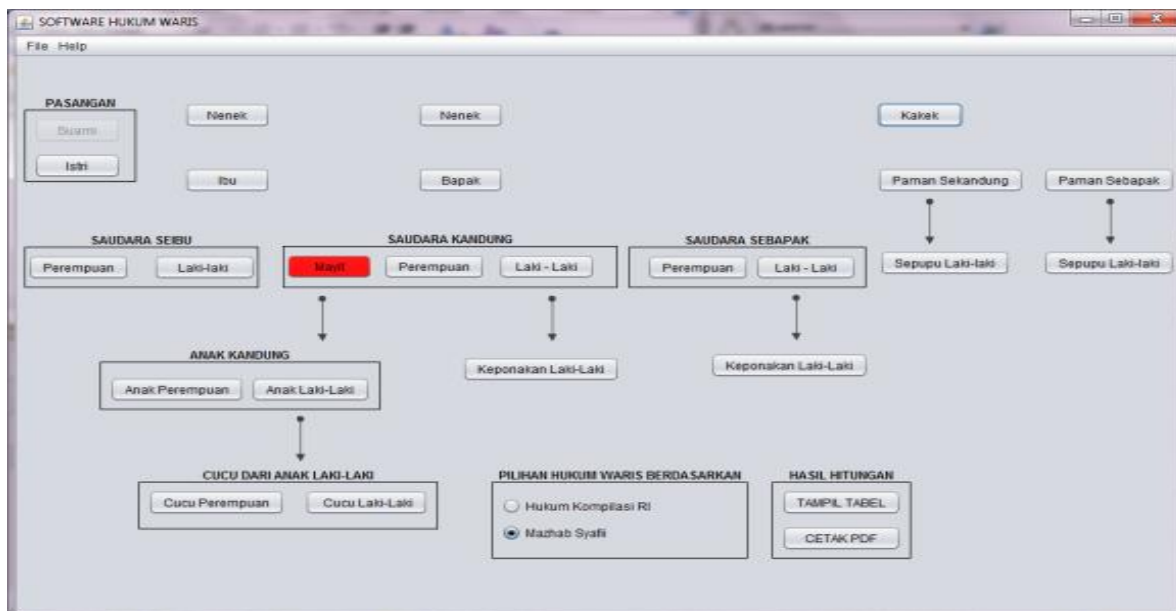
- [1] J. Deacon, *Model-View-Controller (MVC) Architecture*, United Kingdom, 2009.
- [2] S. Dharwiyanti, *Pengantar Unified Modelling (UML)*, Jakarta : ilmukomputer.com, 2003.
- [3] Direktorat Urusan Agama Islam dan Pembinaan Syariah Kementerian Agama Republik Indonesia, *Panduan Praktis Pembagian Waris Dalam Islam*, Kementerian Agama Republik Indonesia, 2011.
- [4] Komite Fakultas Syariah Universitas Al Azhar, Mesir, *Hukum Waris Terlengkap*, Jakarta: Senayan Abadi Publishing, 2011.
- [5] Morales-Chaparro, R., Linaje, M., Preciado, J. C., Sánchez-Figueroa F., *MVC Web design patterns and Rich Internet Applications*, Universidad de Extremadura.
- [6] A. Pillay, *Object Oriented Programming using Java*, University of KwaZulu-Natal, 2007.
- [7] Whitten, Jeffery L., Lonnie D. Bentley., Kevin C. Dittman, *Metode Desain dan Analisis Sistem* Edisi 6, Yogyakarta: Andi, 2004.



Gambar 12. Objek diagram kelompok ahli waris



Gambar 14. Package diagram Model View Controller (MVC)



Gambar 18. Tampilan menu utama