

Prototype Function Flow Diagram (FFD) - Combined Unified Modelling Language with Data Flow Diagram

Rochim Widaryanto^{1*}, Andi Widiyanto².

^{1,2}Teknik Informatika, Universitas Muhammadiyah Magelang

*Email: rochimxwidaryanto@gmail.com

Abstrak

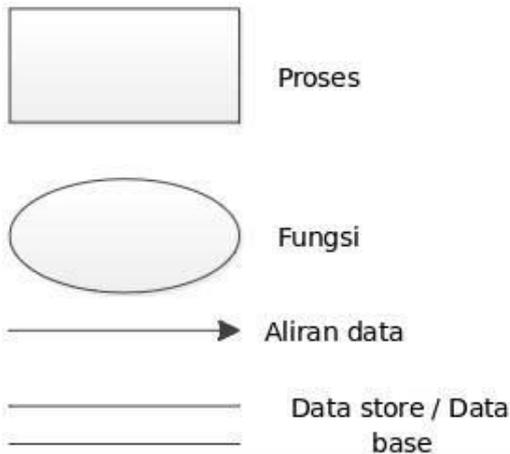
Keywords:
SDLC; UML; DFD;
FFD; Sistem.

System Development Life Cycle (SDLC) adalah proses yang digunakan dalam pembuatan sistem. Object Oriented Programming bertujuan untuk mengombinasikan data dan mengasosiasikan method ke dalam satu unit yang bernama objek untuk membangun sistem yang kompleks.. Belum ada satu model yang efektif untuk semua penerapan, karena itu perlu memilih model yang sesuai untuk setiap project. UML saat ini menjadi tool paling penting untuk sistem dengan model berorientasi objek. Penelitian ini dilakukan dengan melakukan percobaan dari hipotesis yang ada, yaitu sebuah perancangan dengan bentuk yang mirip dengan dfd dengan simbol panah berupa proses atau interaksi yang terjadi di dalam sistem, baik dari user ke sistem maupun sistem ke data, dengan bagan terdiri atas aktor, sistem dan data dan dapat dijabarkan lagi pada level berikutnya seperti pada DFD, dapat digunakan untuk perancangan program berorientasi objek. Hasil kombinasi dari UML dengan bentuk DFD menjadi satu konsep yaitu FFD. FFD dapat dilakukan sampai level dasar yaitu fungsi dan query database sehingga FFD akan mempermudah programmer dalam melihat rancangan sistem.

1. PENDAHULUAN

System Development Life Cycle (SDLC) adalah proses yang digunakan dalam pembuatan sistem. Object Oriented Programming bertujuan untuk mengombinasikan data dan mengasosiasikan method ke dalam satu unit yang bernama objek untuk membangun sistem yang kompleks.. Belum ada satu model yang efektif untuk semua penerapan, karena itu perlu memilih model yang sesuai untuk setiap project. UML saat ini menjadi tool paling penting untuk sistem dengan model berorientasi objek. Penelitian ini dilakukan dengan melakukan percobaan dari hipotesis yang ada, yaitu sebuah perancangan dengan bentuk yang mirip dengan dfd dengan simbol panah berupa proses atau interaksi yang

terjadi di dalam sistem, baik dari user ke sistem maupun sistem ke data, dengan bagan terdiri atas aktor, sistem dan data dan dapat dijabarkan lagi pada level berikutnya seperti pada DFD, dapat digunakan untuk perancangan program berorientasi objek. Hasil kombinasi dari UML dengan bentuk DFD menjadi satu konsep yaitu FFD. FFD dapat dilakukan sampai level dasar yaitu fungsi dan query database sehingga FFD akan mempermudah programmer dalam melihat rancangan sistem.



Gambar 1. Simbol FFD

Untuk menggunakan FFD, perancangan data ERD terlebih dahulu sudah dilakukan. Bentuk perancangan dan pengembangan yang digunakan meniru DFD. Proses yang ada pada sistem dan proses yang terjadi dari interaksi sistem ke database terbentuk menjadi sebuah kelas yang didalamnya terdiri atas fungsi fungsi. Dari perancangan ini, maka FFD dapat digunakan untuk pemrograman berorientasi objek. Bagian ini sebenarnya meniru proses aliran flow dari Use case diagram dan sequence diagram dari UML.

Pada penelitian yang dilakukan, dikombinasikan model flow dari DFD dengan UML sehingga dapat digunakan untuk pemrograman berorientasi objek. Tujuannya ialah untuk memperigkas Diagram. Jika UML memerlukan use case, sequence diagram, di sini didesain dengan diagram FFD.

2. METODE

Hipotesis, sebuah perancangan dengan bentuk yang mirip dengan dfd, dengan simbol panah berupa proses atau interaksi yang terjadi di dalam sistem, baik dari user ke sistem maupun sistem ke data, dengan bagan terdiri atas aktor, sistem dan data dan dapat dijabarkan lagi pada level berikutnya seperti pada DFD, dapat digunakan untuk perancangan program berorientasi objek.

Penelitian ini dilakukan dengan melakukan percobaan dari hipotesis yang ada. Studi literatur dilakukan dengan mengumpulkan data dari jurnal mengenai DFD dan UML.

3. HASIL DAN PEMBAHASAN

3.1. Konsep dasar FFD

Pada UML, Use case diagram menggambarkan interaksi antara sistem dan aktor. Sequence diagram menjelaskan urutan objek / yang harus dilakukan untuk menghasilkan sesuatu.

Untuk menggunakan FFD, perancangan data ERD terlebih dahulu sudah dilakukan. Bentuk perancangan dan pengembangan yang digunakan meniru DFD. Fungsi pada DFD menjadi Proses pada FFD. Aliran data pada DFD menjadi Aliran yang terdiri atas 3 jenis yaitu actor ke proses, proses ke proses dan proses ke database/tabel

FFD level 0 menjabarkan semua pengguna sistem, sistem dan database yang digunakan, acak interaksi yang dilakukan oleh user ke sistem, maupun sistem ke database.

Pengembangan level 0 ke level berikutnya akan menjabarkan Proses yang ada pada sistem dan proses yang terjadi dari interaksi sistem ke database terbentuk menjadi sebuah kelas yang didalamnya terdiri atas fungsi fungsi. Dari perancangan ini, maka FFD dapat digunakan untuk pemrograman berorientasi objek. Bagian ini sebenarnya meniru proses aliran flow dari Use case diagram dan sequence diagram dari UML.

Atribut dari FFD :

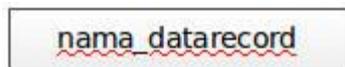
- User (aktor/fungsi) digambarkan dengan logo orang. Hanya yang dapat mengakses sistem yang disebut sebagai user.



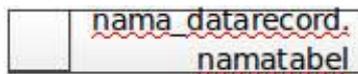
- b. Proses merupakan sebuah sistem. Proses akan dijabarkan lebih detail pada level berikutnya dengan nama P1.1 nama_proses, P1.2 nama_proses, P1.1.1 nama_proses dan seterusnya.



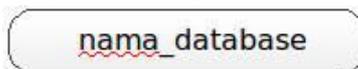
- c. Datarecord merupakan salah satu tempat sistem menyimpan data sementara.. Satu sistem dapat memiliki lebih dari satu datarecord. Datarecord ditunjukkan sebagai Dr1, Dr2.



- d. Data record table merupakan salah satu tempat sistem menyimpan data sementara.. Satu sistem dapat memiliki lebih dari satu datarecord. Pada FFD level berikutnya, tabel dari datastore tertentu ditunjukkan sebagai Dr1.nama_tabel. Dr2.nama_tabel.

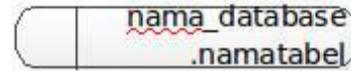


- e. Database merupakan salah satu tempat sistem menyimpan data yang semi permanent (dapat disimpan jangka panjang, dimodifikasi, maupun dihapus). Satu sistem dapat memiliki lebih dari satu database. Pada FFD Database ditunjukkan sebagai Db1. Db2.

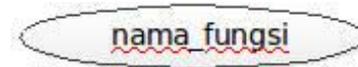


- f. Database table merupakan salah satu tempat sistem menyimpan data yang semi permanent (dapat disimpan jangka panjang, dimodifikasi, maupun dihapus). Satu sistem dapat memiliki lebih dari satu database table. Pada FFD level berikutnya, tabel dari database tertentu

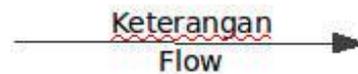
ditunjukkan sebagai Db1.nama_tabel. Db2.nama_tabel.



- g. Fungsi atau prosedur ditandai dengan ellipse.



- h. Flow yang ada pada FFD ada 3 jenis



- Actor to proses : merupakan interaksi atau perintah actor terhadap sistem. Perintah actor berupa kata kerja.
 - Proses to proses : merupakan interaksi internal fungsi/prosedur yang berhubungan dengan fungsi/prosedur lain di dalam sistem.
 - Proses to database/tabel : merupakan interaksi sistem dengan database. Fungsi sistem untuk mengakses database dijabarkan. Hubungan fungsi dengan query yang ada dijabarkan pada tabel proses query.
- i. Tabel proses fungsi berisi detail dari fungsi/prosedur yang terjadi dalam sistem
- j. Tabel proses query berisi detail dari query yang terjadi dari flow fungsi ke database

3.2. Simulasi Proses pembuatan FFD

- a. ERD Dbmarket

Jika FFD menggunakan Data, maka data merupakan hal yang pertama kali dibuat sebelum pembuatan FFD. Tabel Data terbentuk dari ERD.



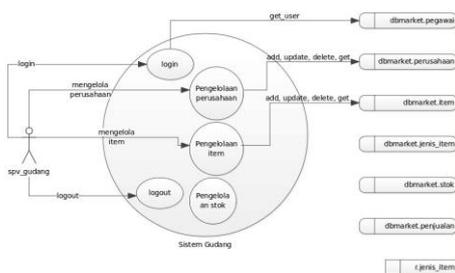
Gambar 2. Tabel data

b. FFD level 0

Pada FFD level 0 dijabarkan apa yang dilakukan oleh user. Jika ada balikan dari sistem berupa tampilan ditulis dengan “show”. “notifikasi” merupakan peringatan/keterangan dari sistem ke user.

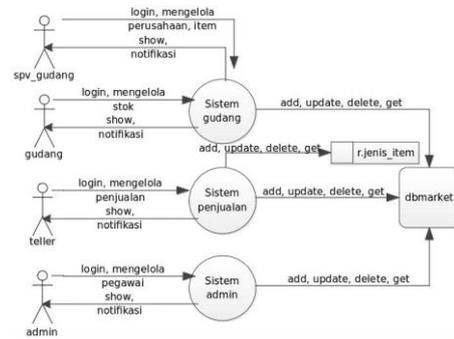
Pada proses internal sistem, terdapat fungsi “set” digunakan untuk mengeset nilai variable, “get” untuk mengambil data dan fungsi fungsi lain dijelaskan pada FFD level berikutnya.

Dari sistem ke datarecord/record/database/table ditulis dengan “add” untuk penambahan data, “update”, “delete” untuk menghapus data dan “get” untuk mengambil data. Jika hal tersebut tidak dilakukan misal tidak diperlukan “update”, maka tidak perlu dituliskan. Penjabaran lebih detail dilakukan pada FFD level berikutnya.



Gambar 3. FFD Level 0

c. FFD level 1 sistem gudang - spv gudang. Pada FFD level 1, dijabarkan per sistem per user. Flow berisi fungsi yang digunakan oleh sistem.



Gambar 4. FFD Level 1

Tabel 1. process sistem gudang - spv gudang

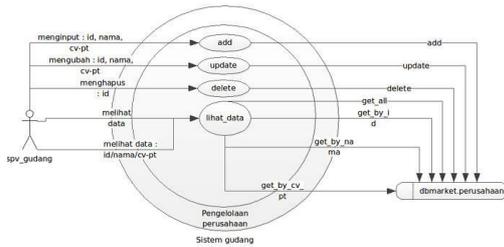
login(nama, pass)	If(login(nama,pass)) sukses maka Pengelolaan perusahaan dan pengelolaan item enable = true If(login(nama,pass)) gagal maka notifikasi “login gagal / input tidak sesuai”
logout()	Pengelolaan perusahaan dan pengelolaan item enable = false: sub aktif (dari mobil 8-12") yang include crossover aktif & switch lpf + speaker pasif full range / coaxial + ampli. Notifikasi logout “sistem deactivated”

Tabel 2. process to table sistem gudang - spv gudang

get_user(nama, pass, bagian)	“select * from pegawai where nama = ‘nama’ and pass = ‘pass’ and bagian = ‘bagian’”
------------------------------	---

Dari tabel di atas dapat diidentifikasi kelas user dengan fungsi-fungsi yang diambil tabel 1 dan tabel 2.

d. FFD level 2 sistem gudang - spv gudang - pengelolaan perusahaan



Gambar 5. FFD Level 2

Tabel 3. process sistem gudang - spv gudang - pengelolaan perusahaan

lihat_data(case)	If(lihat semua data) then get_all else If(lihat data berdasar id) then get_by_id(id) else If(lihat data berdasar nama) then get_by_nama(nama) else(lihat data berdasar cv_pt) then get_by_cv_pt(cv_pt) Notifikasi : Jika id/nama/cv_pt yang diinputkan tidak ditemukan : “data tidak ditemukan” Jika data yang dimasukkan tidak lengkap : “cek kembali data inputan”
add(id, nama, cv_pt)	If (data yang dimasukkan tidak lengkap) notifikasi : “cek kembali data inputan”, else add(id, nama, cv_pt)
update(id, nama, cv_pt)	If(data yang dimasukkan tidak lengkap), notifikasi : “cek kembali data inputan” else if(id yang diinputkan tidak ditemukan), notifikasi : “data tidak ditemukan” else update(id, nama, cv_pt)
delete(id)	If(data yang dimasukkan tidak lengkap), notifikasi : “cek kembali data inputan” else if(id yang diinputkan tidak ditemukan), notifikasi : “data tidak ditemukan” else delete(id)

Tabel 4. process to table sistem gudang - spv gudang - pengelolaan perusahaan

add(id, nama, cv_pt)	“insert into perusahaan values(‘id’, ‘nama’, ‘cv_pt’)”
update(id, nama, cv_pt)	“update perusahaan set nama = ‘nama’, cv_pt = ‘cv_pt’ where id = ‘id’”
delete(id)	“delete perusahaan where id = ‘id’”

get_all	“select * from perusahaan”
get_by_id(id)	“select * from perusahaan where id = ‘id’”
get_by_nama (nama)	“select * from perusahaan where nama = ‘nama’”
get_by_cv_pt (cv_pt)	“select * from perusahaan where cv_pt = ‘cv_pt’”

Dari tabel di atas dapat diidentifikasi kelas pengelolaan_perusahaan dengan fungsi-fungsi yang diambil tabel 3 dan tabel 4.

e. FFD level . . .

FFD dikembangkan sampai tingkat dimana sistem dapat dibaca jelas keseluruhan alur (seluruh fungsi sudah dijabarkan)sehingga pengguna dari FFD maupun programmer dapat membuat program yang sesuai.

4. KESIMPULAN

UML dengan model flow mendekati DFD dapat untuk perancangan software object oriented. FFD dapat dilakukan sampai level dasar yaitu fungsi/prosedur dan query database sehingga FFD akan mempermudah programmer dalam melihat sistem.

REFERENSI

[1] Kirti Tiwari, Alpika Tripathi, Shipra Sharma, Vandana Dubey. Merging of Data Flow Diagram with Unified Modeling Language. International Journal of Scientific and Research Publications2 2012; Volume 2, Issue 8.

[2] M.L.V.Roopaa Vani, M.Chandrika Kumari, M.Hari Priya, N.Harika. An Effective Language for Object Oriented Design-UML (Unified Modeling Language). International Research Journal of Engineering and Technology (IRJET);2012; Volume: 02 Issue: 05.

[3] Munassar, N. ; Govardhan, A. Comparison between Traditional Approach and Object-

- Oriented Approach in Software Engineering 126 Development. International Journal of Advanced Computer Science and Applications(IJSCA);2011; Vol. 2, No. 6. [4] Rosziati Ibrahim, Siow Yen Yen. A Formal Model for Data Flow Diagram Rules, AJSS Journal;2012; Volume 1 No. 2.