

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

Rochim Widaryanto^{1*}, Andi Widiyanto², Affan Rifa'i³

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

³Teknik Industri, Universitas Muhammadiyah Magelang

Email: ^{1}rochimxwidaryanto@gmail.com, ²andi.widiyanto@ummgl.ac.id, ²affan_r@engineer.com

Abstrak

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

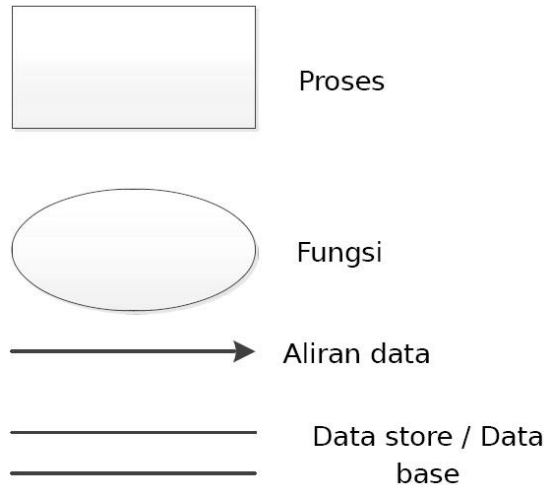
System Development Life Cycle (SDLC) adalah proses yang digunakan dalam pembuatan sistem. Object Oriented Programming bertujuan untuk mengombinasikan data dan mengasosiasikan method ke dalam satu unit yang bernama objek untuk membangun sistem yang kompleks. Belum ada satu model yang efektif untuk semua 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. Ada 4 fase pada SDLC, yaitu Planning, Analisis, Desain, Implementasi [4]. Beberapa model telah diciptakan. Belum ada satu model yang efektif untuk semua penerapan, karena itu perlu memilih model yang sesuai untuk setiap project [2].

Object Oriented Programming bertujuan untuk mengombinasikan data dan mengasosiasikan method ke dalam satu unit yang bernama objek untuk membangun sistem yang kompleks. UML merupakan sekumpulan diagram dengan aspek yang berbeda yang digunakan untuk memodelkan software berorientasi objek. UML saat ini menjadi tool paling penting untuk sistem dengan model berorientasi objek [3].

DFD memiliki representasi node dan aliran data. Node dapat berupa data store, auxiliary store, proses, atau terminator. Panah berhubungan dengan aliran data. Node dan aliran data harus diberi label yang menandakan node ataupun aliran data tersebut [1].



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.

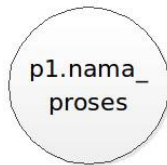
Atribut dari FFD :

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



actor

- b. Proses merupakan sebuah sistem. Proses dijabarkan dengan nama P1.nama proses, P2 nama proses dan seterusnya.



- c. Kelas ditandai dengan lingkaran, merupakan detail dari isi proses. kelas dijabarkan dengan nama P1.1. nama kelas, P1.2. nama kelas dan seterusnya Fungsi atau prosedur ditandai dengan elipse. fungsi dijabarkan dengan nama P1.1.1. nama fungsi, P1.1.2. nama fungsi dan seterusnya
- d. Fungsi atau prosedur ditandai dengan elipse. fungsi dijabarkan dengan nama P1.1.1. nama fungsi, P1.1.2. nama fungsi dan seterusnya



- e. Fungsi atau prosedur ditandai dengan elipse.

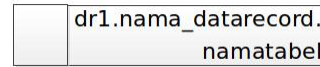


- f. Datarecord merupakan salah satu tempat sistem menyimpan data sementara. Satu sistem dapat memiliki lebih dari satu datarecord. Datarecord ditunjukkan sebagai Dr1, Dr2.

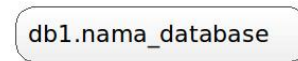


- g. Datarecord namatable merupakan salah satu tempat sistem menyimpan data sementara. Satu

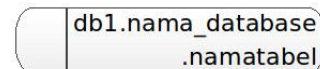
sistem dapat memiliki lebih dari satu datarecord. Pada FFD level berikutnya, tabel dari datarecord tertentu ditunjukkan sebagai Dr1.nama_tabel. Dr2.nama_tabel.



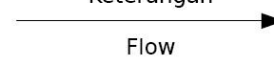
- h. Database merupakan salah satu tempat sistem menyimpan data yang semi permanent (dapat disimpan jangka panjang, dimodifikasi, maupun dihapus). Satu sistem dapat memiliki lebih dari satu database. Pada FFD Database ditunjukkan sebagai Db1. Db2.



- i. Database namatable 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 namatable. Pada FFD level berikutnya, tabel dari database tertentu ditunjukkan sebagai Db1.nama_tabel. Db2.nama_tabel.



- j. Flow yang ada pada FFD ada 3 jenis Keterangan



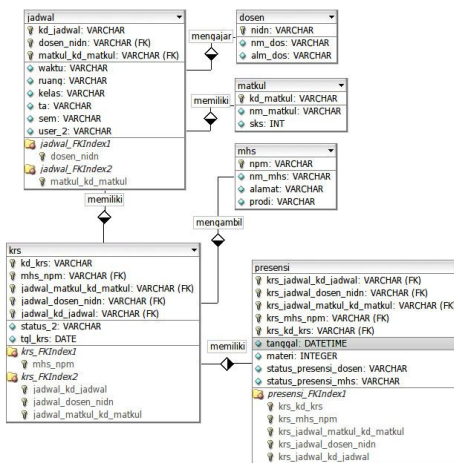
- Actor to proses : merupakan interaksi atau perintah actor terhadap sistem. Perintah actor berupa kata kerja.
- Proses to proses : merupakan interaksi internal fungsi/prosedur yang berhubungan dengan fungsi/prosedur lain di dalam sistem.
- Proses to database/tabel : merupakan interaksi sistem dengan database. Fungsi sistem untuk mengakses database dijabarkan. Hubungan fungsi dengan query yang ada

- dijabarkan pada tabel proses query.
- k. Tabel proses fungsi berisi detail dari fungsi/prosedur yang terjadi dalam sistem
 - l. Tabel proses query berisi detail dari query yang terjadi dari flow fungsi ke database

3.2. Simulasi Proses pembuatan FFD

1. ERD Dbpresensi

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



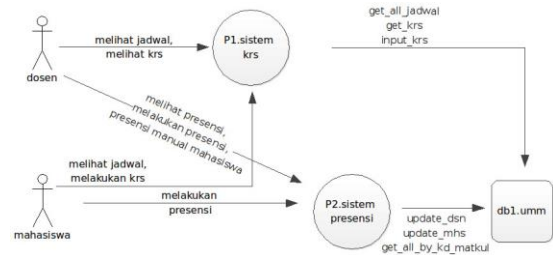
Gambar 2. Tabel data

2. 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.

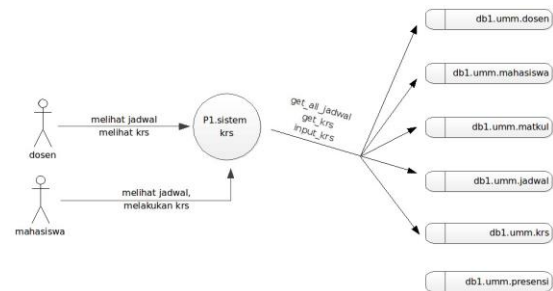
Pada proses internal sistem, terdapat fungsi “set” digunakan untuk mengeset nilai variable, “get” untuk mengambil data dan fungsi fungsi lain dijelaskan pada FFD level berikutnya.

Dari sistem ke datarecord/database ditulis dengan fungsi yang akan digunakan. Penjabaran lebih detail dilakukan pada FFD level berikutnya.

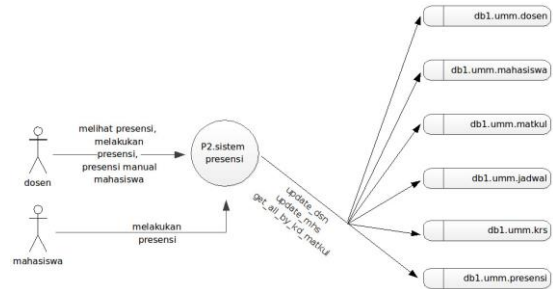


Gambar 3. FFD Level 0

3. FFD level 1 sistem krs dan sistem presensi. Pada FFD level 1, dijabarkan per sistem dan memunculkan tabel dari database.

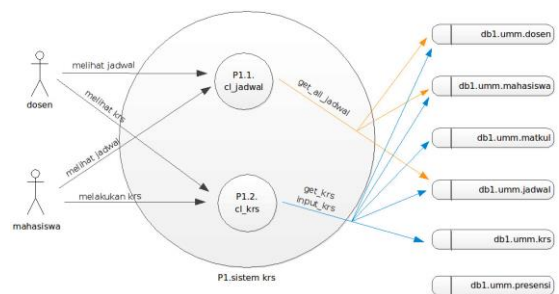


Gambar 4. FFD Level 1 Sistem Krs

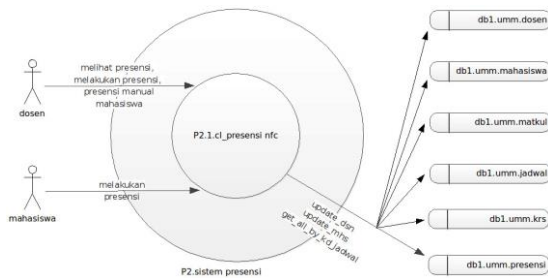


Gambar 5. FFD Level 1 Sistem Presensi

4. FFD level 2 sistem krs dan sistem presensi. kelas yang digunakan oleh sistem dimunculkan.



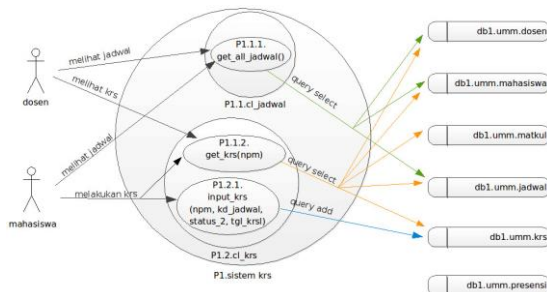
Gambar 6. FFD Level 2 Sistem Krs



Gambar 7. FFD Level 2 Sistem Presensi

	melihat jadwal
dosen	1.1.2. memilih untuk melihat krs
mahasiswa	1.1.2. memilih untuk melihat jadwal
mahasiswa	1.2.1. memasukkan kd_jadwal

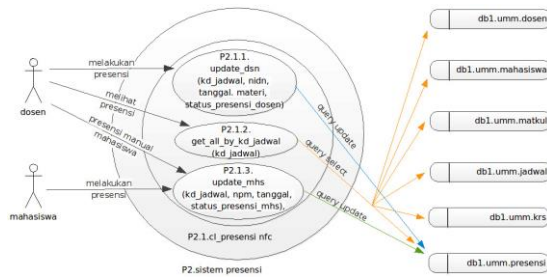
5. FFD level 3 sistem krs dan sistem presensi. fungsi yang digunakan oleh setiap kelas dalam sistem dimunculkan. flow antara fungsi ke database diberi keterangan “query select” untuk mengambil data, “query add” untuk menambah data, “query update” untuk mengubah data atau “query delete” untuk menghapus data



Gambar 8. FFD Level 3 Sistem Krs

Tabel 2. sistem krs pada tabel berikut tidak ada perhitungan di setiap fungsi. jika ada, dijelaskan dengan alur code

fungsi	penjelasan fungsi
1.1.1. get_all_jadwa lQ	tidak ada perhitungan di dalamnya.
1.1.2. get_krs(npm)	tidak ada perhitungan di dalamnya
1.2.1. input_krs(np m, kd_jadwal, status, tgl_krs)	tidak ada perhitungan di dalamnya



Gambar 9. FFD Level 3 Sistem Presensi

Tabel 3. sistem krs - database

fungsi	query
1.1.1. get_all_jadwa lQ	<code>select * from jadwal inner join dosen on jadwal.dosen_nidn = dosen.nidn inner join matkul on jadwal.matkul_kd_matkul = matkul.kd_matkul</code>
1.1.2. get_krs(npm)	<code>select * from krs inner join mhs on krs.mhs_npm=mhs.npm inner join jadwal on krs.jadwal_kd_jadwal=jadwal.kd_jadwal inner join dosen on krs.jadwal_dosen_nidn = dosen.nidn inner join matkul on krs.jadwal_matkul_kd_matk</code>

bagian akhir itu didefinisikan variabel yang diinputkan user dan query yang dilakukan sistem ke database

Tabel 1. user-sistem krs

user	alur
dosen	1.1.1. memilih untuk

	ul = matkul.kd_matkul where mhs_npm = npm
1.2.1. input_krs(npm, kd_jadwal, status, tgl_krs)	insert into krs values input_kd_krs, input_npm, input_kd_matkul, input_nidn, input_kd_jadwal, input_status_2, input_tgl_krs

Tabel 4. user-sistem presensi

user	alur
dosen	2.1.1. melakukan presensi dengan kartu nfc, sistem membaca data dari kartu
dosen	2.1.2. memilih untuk melihat presensi
dosen	2.1.3. memasukkan kd_jadwal, npm dan status presensi dari mahasiswa
mahasiswa	2.1.3. melakukan presensi dengan kartu nfc, sistem membaca data dari kartu

Tabel 5. sistem presensi
pada tabel berikut tidak ada perhitungan di setiap fungsi. jika ada, dijelaskan dengan alur code

fungsi	penjelasan fungsi
2.1.1. update_dsn(kd_jadwal, nidn, tanggal, materi, status_presensi_dosen)	tidak ada perhitungan di dalamnya
2.1.2. get_all_by_kd_jadwal(kd_jadwal)	tidak ada perhitungan di dalamnya

2.1.3.update_dsn(kd_jadwal, npm, tanggal, status_presensi_mahasiswa)	tidak ada perhitungan di dalamnya
--	-----------------------------------

Tabel 6. sistem presensi - database

fungsi	query
2.1.1. update_dsn(kd_jadwal, npm, tanggal, materi, status_presensi_dosen)	update presensi set tanggal = input_tanggal, materi = input_materi, status_presensi_dosen = input_status_presensi_dosen where krs_jadwal_kd_jadwal = kd_jadwal
2.1.2. get_all_by_kd_jadwal(kd_jadwal)	select * from presensi inner join krs on presensi.krs_kd_krs = krs.kd_krs inner join mhs on presensi.krs_mhs_npm=mh s.npm inner join jadwal on krs.jadwal_kd_jadwal=jadw al.kd_jadwal inner join dosen on presensi.krs_jadwal_dosen_ nidn = dosen.nidn inner join matkul on presensi.krs_jadwal_matkul_ kd_matkul = matkul.kd_matkul where krs_jadwal_kd_jadwal = kd_jadwal
2.1.3.update_dsn(kd_jadwal, npm, tanggal, status_presensi_mahasiswa)	update presensi set status_presensi_mahasiswa = input_status_presensi_maha siswa where krs_jadwal_kd_jadwal = kd_jadwal

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

Jurnal, Bulletin, dan Majalah Ilmiah

- [1] Kirti Tiwari, Alpika Tripathi, Shipra Sharma, Vandana Dubey. Merging of Data Flow Diagram with Unified Modeling Language. International Journal of Scientific and Research Publications 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.