

APLIKASI PERHITUNGAN PROFIL ALIRAN DENGAN METODE INTEGRASI GRAFIS DAN TAHAPAN LANGSUNG PADA SALURAN BERPENAMPANG TRAPESIUM

Kanjalia Tjandrapuspa T¹, Adelia², Heri Santoso³

¹ Dosen tetap Jurusan Teknik Sipil, Universitas Kristen Maranatha Bandung

² Dosen tetap Sistem Informasi, Universitas Kristen Maranatha Bandung

³ Alumnus Double Degree, JTS-SI., Universitas Kristen Maranatha Bandung

ABSTRACT

In a dam project must be required calculations used to aid development. This calculation is a calculation of the flow profile with a graphical integration method and direct step method. Purposes of calculating the flow profile to assist in the construction of a dam that river flows well into the weir. Purpose of making this application to simplify the calculations manually. This application is created using the Java programming language with MySQL database. Source of data obtained from the literature and application testing by comparing the calculation and application of manual calculations.

Keywords: flow profile, projects, applications, data.

ABSTRAK

Dalam suatu proyek bendungan pasti dibutuhkan perhitungan yang digunakan untuk membantu pembangunan. Perhitungan yang dibuat ini merupakan perhitungan profil aliran dengan metoda integrasi grafis dan metoda tahapan langsung. Tujuan perhitungan profil aliran untuk membantu dalam pembangunan bendungan agar aliran sungai mengalir dengan baik menuju bendung. Tujuan dari pembuatan aplikasi ini untuk mempermudah perhitungan secara manual. Aplikasi yang dibuat ini menggunakan bahasa pemrograman Java dengan *database* MySQL. Sumber data yang diperoleh dari literatur dan pengujian aplikasi dengan membandingkan perhitungan aplikasi dan perhitungan secara manual.

Kata kunci: profil aliran, proyek, aplikasi, data.

1. PENDAHULUAN

Sungai memiliki peranan yang sangat penting bagi kehidupan manusia. Hal ini dapat dilihat dari pemanfaatan sungai yang makin lama makin kompleks, mulai dari sarana transportasi, sumber air baku, sumber tenaga listrik dan sebagainya. Sungai yang dimanfaatkan untuk sumber tenaga listrik akan membuat bendungan untuk mengaliri air dari sungai menuju mesin pembangkit listrik.

Dengan adanya bendung maka permukaan air akan berubah lambat laun akan membentuk suatu profil aliran. Tujuan perhitungan profil aliran untuk membantu dalam pembangunan atau perancangan bendungan agar aliran sungai mengalir dengan baik menuju bendung. Bentuk penampang saluran yang umum dipakai untuk saluran adalah trapesium karena stabilitas kemiringan dindingnya dapat disesuaikan.

Rumusan Masalah:

1. Bagaimana membuat aplikasi perhitungan profil aliran dengan menggunakan metoda integrasi grafis dan tahapan langsung pada saluran berpenampang trapesium?
2. Bagaimana perhitungan manual dibandingkan dengan hasil dari aplikasi?

Tujuannya adalah:

1. Untuk mengetahui cara aplikasi perhitungan profil aliran dengan metoda integrasi grafis dan tahapan langsung pada saluran berpenampang saluran
2. Untuk mengetahui hasil perhitungan manual dibandingkan dengan hasil dari aplikasi

2. KAJIAN TEORI

Hidrolika

Hidrolika adalah bagian dari hidrodinamika yang berhubungan dengan gerak air atau mekanika aliran. Ditinjau dari mekanika aliran, terdapat dua macam aliran yaitu aliran saluran tertutup dan aliran saluran terbuka. Dua macam aliran tersebut dalam banyak hal mempunyai kesamaan tetapi memiliki perbedaan juga. Perbedaan tersebut adalah pada keberadaan permukaan bebas; aliran saluran terbuka mempunyai permukaan bebas, sedangkan aliran saluran tertutup tidak mempunyai permukaan bebas karena air mengisi seluruh penampang saluran (Triatmodjo, 1993).

Aliran saluran terbuka (*open channel flow*) sangat erat hubungannya dengan teknik sipil. Seperti yang harus diketahui, air mengalir dari hulu ke hilir sampai mencapai suatu elevasi permukaan air tertentu. Kecenderungan ini ditunjukkan oleh aliran di saluran alam yaitu sungai. Perjalanan air dapat juga ditambah oleh bangunan-bangunan yang dibuat oleh manusia seperti saluran irigasi, pipa, gorong-gorong (*culvert*), bendungan, dan saluran buatan yang lain atau kanal.

Elemen Geometri

Penampang saluran (*channel cross section*) adalah penampang yang diambil tegak lurus arah aliran, sedang penampang yang diambil vertikal adalah penampang vertikal (*vertical section*) (Chow, 1992).

Bentuk penampang trapesium adalah bentuk yang biasa digunakan untuk saluran-saluran irigasi atau saluran-saluran drainase karena menyerupai bentuk saluran alam, dimana kemiringan tebingnya menyesuaikan dengan sudut lereng alam dari yang digunakan untuk saluran alam. Bentuk penampang persegi empat atau segitiga merupakan penyederhanaan dari bentuk trapesium yang biasanya digunakan untuk saluran-saluran

drainase yang melalui lahan-lahan yang sempit. Bentuk penampang lingkaran biasanya digunakan pada perlintasan dengan jalan. Saluran ini disebut gorong-gorong (*culvert*).

Profil Aliran

Profil aliran adalah bentuk dari aliran yang menuju bendung atau terhambat oleh bangunan air atau irigasi. Tujuan perhitungan profil aliran untuk mengetahui lokasi pembendungan. Profil aliran dapat digolongkan berdasarkan sifat kemiringan saluran dan zona tempat permukaan aliran. Perhitungan profil aliran berubah lambat laun (*gradually varied flow*) pada dasarnya meliputi penyelesaian persamaan dinamis dari aliran berubah lambat laun. Sasaran utama dari perhitungan adalah menentukan bentuk profil aliran. Ada tiga metoda perhitungan yaitu metoda integrasi grafis (*graphical integration method*), metoda integrasi langsung (*direct integration method*), dan metoda tahapan langsung (*step method*) (Chow, 1992).

Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem. *Use case diagram* menggambarkan kebutuhan sistem dari sudut pandang *user*, memfokuskan pada proses komputerisasi dan menggambarkan hubungan antara *use case* dan *actor*. Fitur-fitur yang ada di dalam sistem dapat digunakan untuk membantu mengelompokkan sebuah sistem, yang digunakan UML untuk menggambarkan bagaimana *actor* menggunakan sistem tersebut. Komponen pembentuk *Use case diagram* adalah :

a. Actor

Actor menggambarkan sebuah tugas/peran yang digunakan dalam mengakses sistem. *Actor* dapat berupa manusia, sistem, atau *device* yang menyediakan atau menerima informasi dari sistem. *Actor* biasanya menggunakan kata benda. Tidak boleh ada komunikasi langsung antar *Actor*. Satu *Actor* dapat menggunakan banyak *use case*, sebaliknya juga satu *use case* dapat digunakan oleh lebih dari satu *Actor*.

b. Use Case

Use case dapat dianggap sebagai sebuah perilaku dari suatu sistem yang membantu sebuah *actor* untuk mencapai tujuan melalui sistem ini. *Use case* menggambarkan *goal* yang harus dicapai oleh sistem. Suatu *use case* difokuskan pada *goal* yang ingin dicapai, bukan bagaimana mengimplementasikan *goal* tersebut. *Use case* biasanya menggunakan

kata kerja. Nama *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.

c. Relationship

Relationship merupakan penghubung antara *actor* dengan *use case*. Ada tiga jenis *relationship*, yaitu:

1. Asosiasi, mengidentifikasi interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu. Tiap asosiasi menjadi dialog yang harus dijelaskan dalam *use case narrative*. Asosiasi bisa berarah (garis dengan anak panah) jika komunikasi satu arah dan dapat juga tidak berarah jika komunikasi terjadi dua arah.
2. Generalisasi, untuk mendefinisikan relasi antara dua *actor* atau dua *use case* yang mana salah satunya menurunkan dan menambahkan sifat dari yang lainnya. Digambarkan menggunakan garis bermata panah kosong dari turunannya mengarah ke *parent* nya.
3. Dependensi, terdiri dari dua jenis yaitu:
 - a. *Include*, mengidentifikasi hubungan antar dua *use case* dimana yang satu memanggil yang lain. Digambarkan dengan garis putus-putus bermata panah dengan notasi <<*include*>> pada garis. Dimana setelah *use case* utama dijalankan, *sub use case* ini pun dijalankan. Tanda panah terbuka harus terarah ke sub *use case*.
 - b. *Extend*, terjadi jika *pemanggilan* memerlukan adanya kondisi tertentu. Digambarkan serupa dengan dependensi <<*include*>> kecuali arah panah berlawanan dan kata “*include*” diganti dengan “*extend*”. Biasanya kondisi tersebut adalah ketika seorang *Actor* mengakses *sub use case* ini.

Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek (*Chonoles dan Schardt, 2003*). *Class* menggambarkan keadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*method/function*).

Terdapat beberapa hubungan antar *class* pada *class diagram*, yaitu (Ambler, 2005):

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*. Banyaknya asosiasi diberi label pada kedua ujung baris, satu jumlah besar indikator untuk masing-masing arah.

2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambahkan fungsionalitas baru, sehingga disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan melalui *sequence diagram* untuk mengetahui nilai apa saja yang diterima sistem dan yang dikirimkan sistem kepada *Actor*.

Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku *Use Case* dan *objects* di dalam sistem. *Activity diagram* merupakan teknik untuk menjelaskan proses bisnis, procedural logika, dan alur kerja (Chonoles dan Schardt, 2003). Diagram ini dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas proses bisnis. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*. *Activity diagram* tidak menggambarkan *behavior* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Entity Relationship Diagram (ERD)

Entity Relationship Diagram merupakan sarana komunikasi antara perancang basis data dan pengguna sistem selama tahap analisis dari proses pengembangan basis data dalam kerangka pengembangan sistem informasi. Digunakan untuk mengkonstruksi model data konseptual, yang mencerminkan struktur dan batasan dari basis data, yang mandiri dari perangkat lunak pengelola basis data dan berhubungan erat dengan model data yang langsung bisa digunakan untuk mengimplementasikan basis data secara logika maupun secara fisik dengan *Database Management System (DBMS)* yang dipilih saat tahapan Implementasi kelak. (Nugroho, Adi. 2004:47-48).

Terdapat beberapa komponen utama pembentuk *Entity Relationship Model*, yaitu (Fathansyah, 2007:73):

1. Entitas

Entitas merupakan sesuatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan.

2. Relasi

Relasi menunjukkan adanya hubungan yang terjadi antara satu atau lebih entitas. Relasi tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara entitas tersebut.

3. Atribut

Atribut merupakan karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Terdapat dua jenis atribut, yaitu:

- a. *Identifer (key)*, untuk menentukan entitas secara unik, dan
- b. *Descriptor (nonkey attribute)*, untuk menentukan karakteristik dari suatu entitas yang tidak unik.

Simbol yang digunakan adalah oval.

Atribut memiliki beberapa tipe, yaitu:

- a. Atribut komposit
Atribut yang dapat dipecah menjadi atribut-atribut lainnya.
- b. Atribut bernilai banyak
- c. Atribut turunan
Atribut yang nilainya bisa didapatkan dari atribut yang lainnya.

4. Cardinality Ratio

Cardinality ratio menjelaskan hubungan batasan jumlah keterhubungan suatu entitas dengan entitas lain atau banyaknya entitas yang bersesuaian dengan entitas lain melalui relasi. Jenis *cardinality ratio*:

- a. *One to One (1 : 1)*
Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B. Begitu juga sebaliknya.
- b. *One to Many (1 : N)* atau *Many to One (M : 1)*
Setiap entitas A dapat berhubungan dengan satu atau lebih entitas B, tetapi entitas B hanya berhubungan minimal satu dengan entitas A.
- c. *Many to Many (M : N)*
Setiap entitas A dapat berhubungan dengan banyak entitas B, dan begitu juga sebaliknya. Setiap entitas B dapat berhubungan dengan satu atau lebih entitas A.

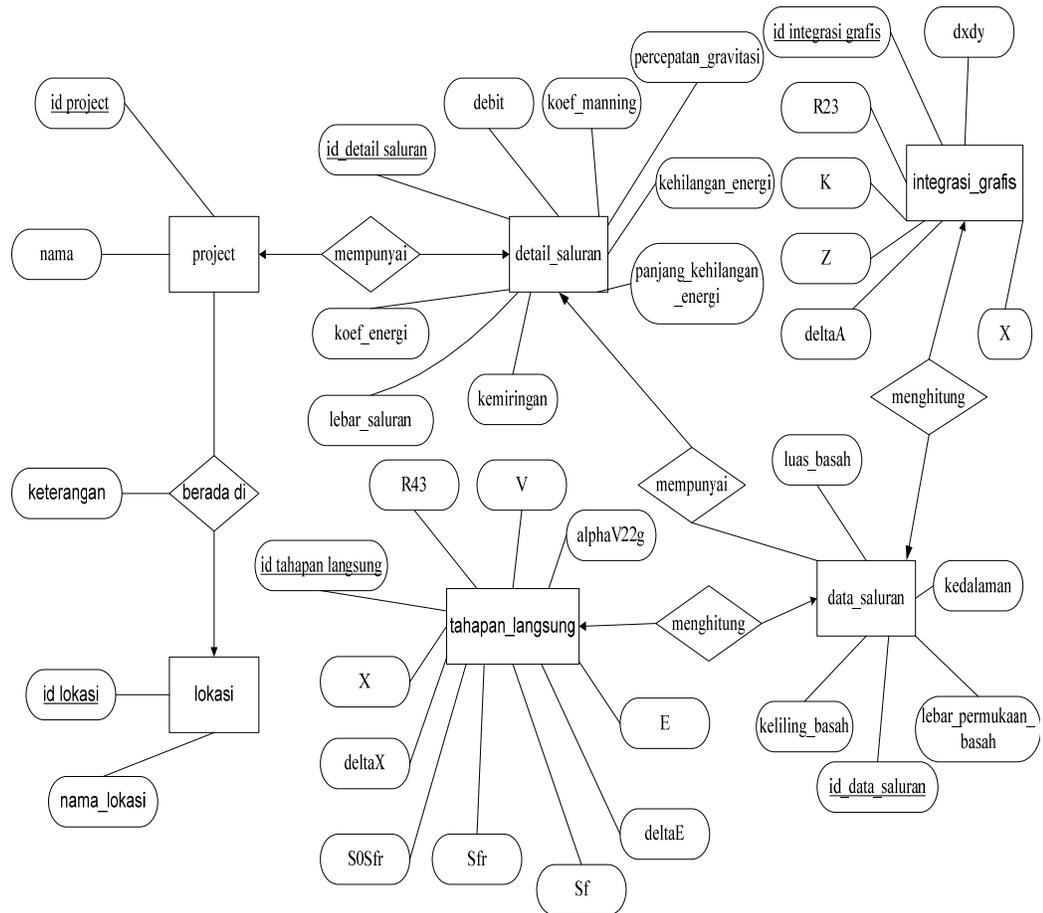
3. ANALISIS DAN PERANCANGAN SISTEM

Proses Bisnis

Proses bisnis dari Sistem Informasi Profil Aliran yaitu :

1. Kontraktor melakukan tinjauan lapangan.
2. Setelah meninjau lapangan, Kontraktor akan mengambil data dari lapangan yang berupa kedalaman normal, debit, lebar saluran dan lebar permukaan basah
3. Kontraktor menghitung profil aliran dari data yang didapatkan dari lapangan.
4. Dari perhitungan, didapatkan hasil berupa yang jarak antara kedalaman yang diinginkan.
5. Kontraktor dapat menganalisa rancangan bangunan bendung yang dibuat dari hasil perhitungan yang didapatkan.

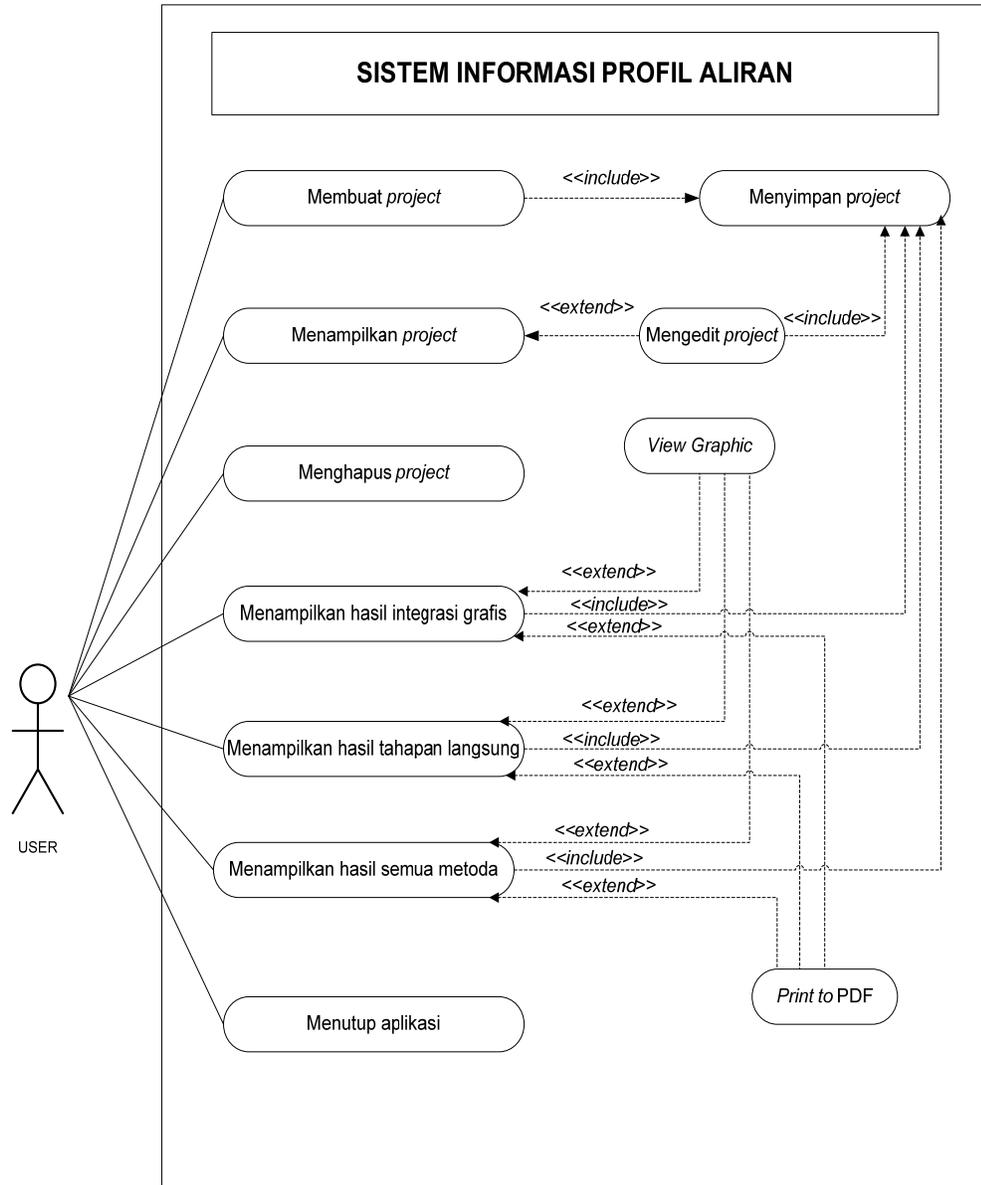
Entity Relationship Diagram (ERD)



Gambar 1. Entity Relationship Diagram.

Use Case

Use Case Analisis Parkir



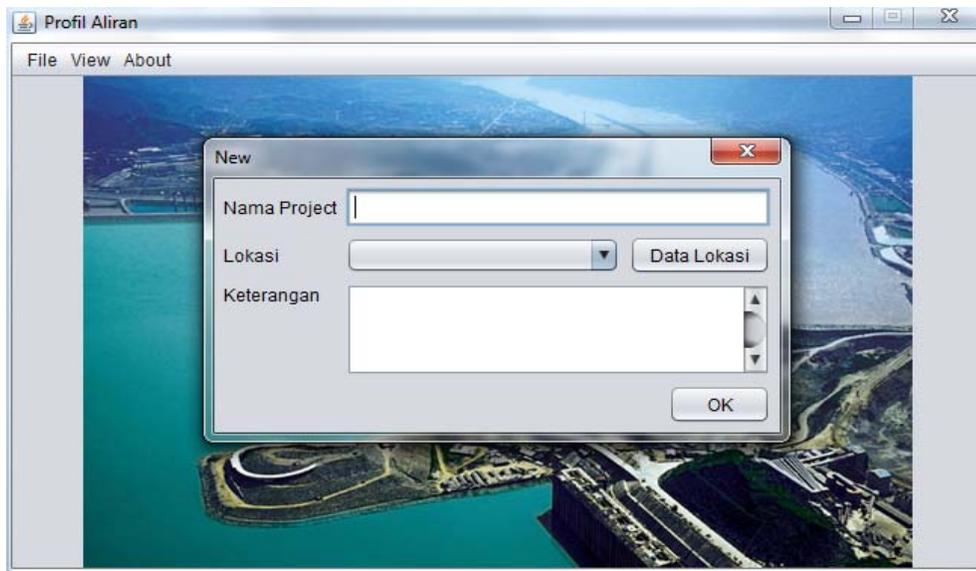
Gambar 2. Use Case Sistem Informasi Profil Aliran.

4. HASIL PENELITIAN



Gambar 3. Tampilan Awal.

Tampilan awal pada saat aplikasi dijalankan dapat dilihat pada Gambar 3. Pada tampilan ini ditampilkan *menu bar File, View dan About*. Pada *menu bar File* ditampilkan *menu item New Project, Open Project, dan Quit* seperti yang dapat dilihat pada Gambar 33.

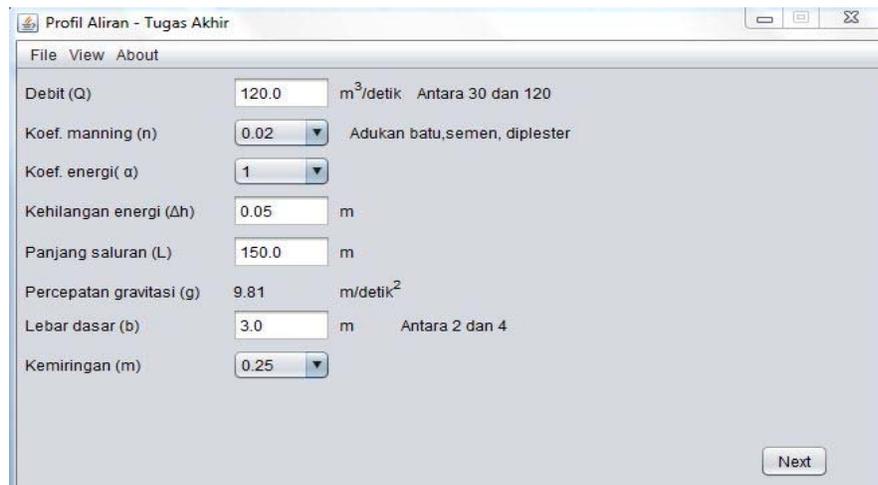


Gambar 4. Tampilan *New Project*.



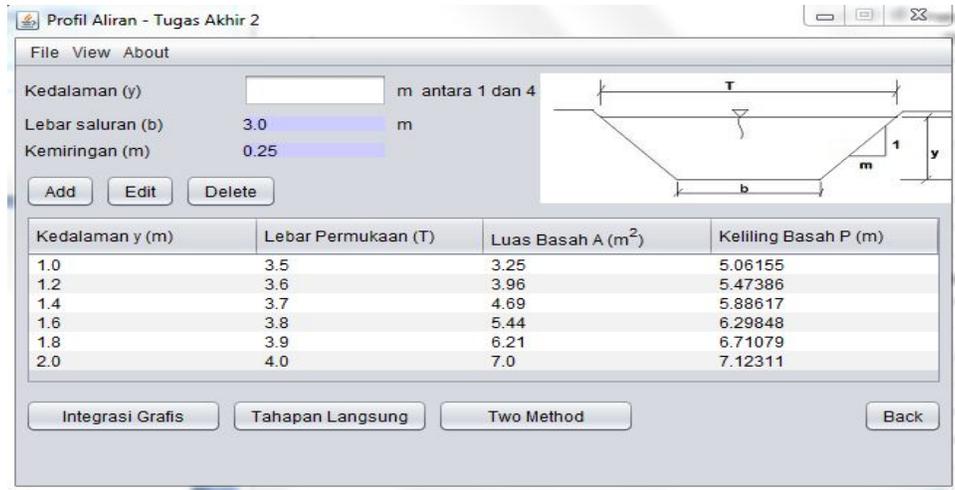
Gambar 5. Tampilan Combo Box New Project.

Jika pengguna memilih menu *New Project*, maka akan muncul tampilan seperti Gambar 4. Di dalam menu *New Project* terdapat *combo box* seperti pada Gambar 5.



Gambar 6. Tampilan Detail Saluran.

Jika pengguna memilih tombol OK pada tampilan *New Project*, maka akan muncul tampilan Detail Saluran seperti Gambar 6. Pengguna memasukkan data yang ada, jika semua data telah dimasukkan dengan benar maka tekan tombol *Next* untuk ke tampilan berikutnya. Jika salah memasukkan data ketika ditekan tombol *Next* akan muncul pesan peringatan.



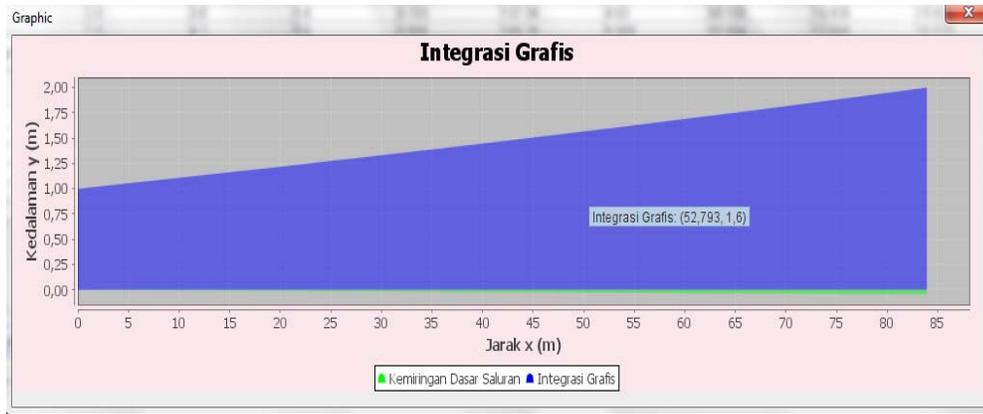
Gambar 7. Tampilan Data Saluran.

Jika pengguna telah memasukkan data dengan benar dan memilih tombol *Next* pada tampilan Detail Saluran, maka akan muncul tampilan Data Saluran seperti Gambar 7. Pengguna bisa menambahkan data dengan tombol *Add*, mengubah data dengan tombol *Edit*, dan menghapus data dengan tombol *Delete*.

y (m)	T (m)	A (m ²)	P (m)	$R^{2/3}$	K	Z	dv/dy	ΔA	x
1.0	2.0	3.0	5.0	0.711	106.65	3.674	94.151	0.0	0.0
1.2	2.0	3.6	5.4	0.763	137.34	4.83	90.106	18.426	18.426
1.4	2.0	4.2	5.8	0.806	169.26	6.086	85.884	17.599	36.025
1.6	2.0	4.8	6.2	0.843	202.32	7.436	81.797	16.768	52.793
1.8	2.0	5.4	6.6	0.875	236.25	8.873	77.837	15.963	68.756
2.0	2.0	6.0	7.0	0.902	270.6	10.392	73.853	15.169	83.925

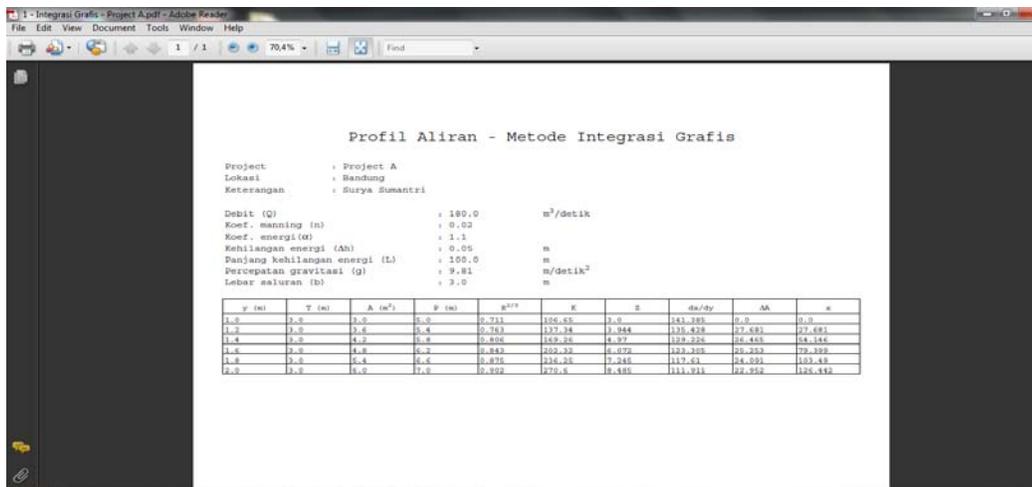
Gambar 8. Tampilan Tabel Integrasi Grafis.

Jika pengguna memilih tombol Integrasi grafis pada Gambar 7, maka akan muncul tampilan seperti Gambar 8.



Gambar 9. Tampilan Grafik Integrasi Grafis.

Jika pengguna memilih tombol *View Graphic* pada Gambar 8, maka akan muncul tampilan seperti Gambar 9.



Gambar 10. Tampilan PDF Integrasi Grafis.

Jika pengguna memilih tombol *Print to PDF* pada Gambar 8, maka akan muncul tampilan seperti Gambar 10.

y (m)	A (m ²)	P (m)	R ^{4/3}	V	Sf	$\alpha \cdot V^2/2g$	E	ΔE	Sfr	S ₀ -Sfr	Δx	x
1.0	3.0	5.0	0.506	60.0	2.84585	201.835	202.835	0.0	0.0	0.0	0.0	0.0
1.2	3.6	5.4	0.582	50.0	1.718213	140.163	141.363	-61.472	2.282032	-2.281532	26.943	26.943
1.4	4.2	5.8	0.65	42.857	1.1302906	102.976	104.376	-36.987	1.424252	-1.423752	25.979	52.922
1.6	4.8	6.2	0.711	37.5	0.7911392	78.842	80.442	-23.934	0.960715	-0.960215	24.926	77.848
1.8	5.4	6.6	0.765	33.333	0.5809616	62.293	64.093	-16.349	0.68605	-0.68555	23.848	101.696
2.0	6.0	7.0	0.814	30.0	0.4422605	50.459	52.459	-11.634	0.511611	-0.511111	22.762	124.458

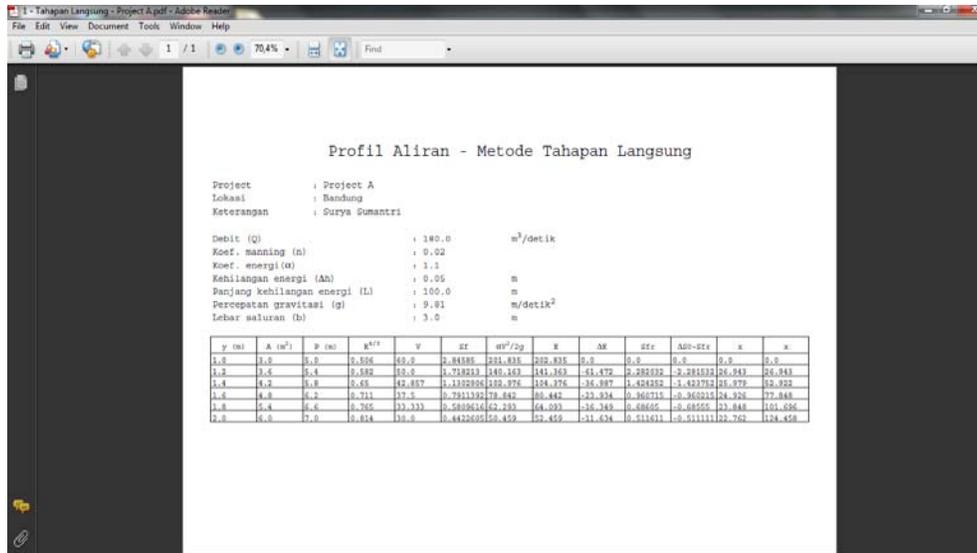
Gambar 11. Tampilan Tabel Tahapan Langsung.

Jika pengguna tombol Tahapan Langsung pada Gambar 7, maka akan muncul tampilan seperti Gambar 11.



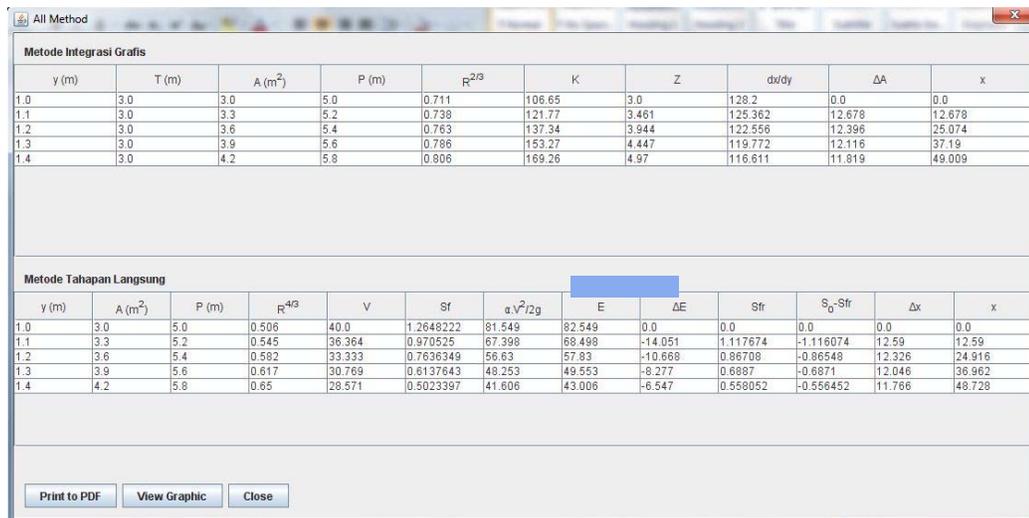
Gambar 12. Tampilan Grafik Tahapan Langsung.

Jika pengguna memilih menu tombol *View Graphic* pada Gambar 11, maka akan muncul tampilan seperti Gambar 12.



Gambar 13. Tampilan PDF Tahapan Langsung.

Jika pengguna memilih tombol *Print to PDF* pada Gambar 11, maka akan muncul tampilan seperti Gambar 13.



Gambar 14. Tampilan Tabel Integrasi Grafis dan Tahapan Langsung.

Jika pengguna memilih tombol *Two Method* pada Gambar 7, maka akan muncul tampilan seperti Gambar 14.



Gambar 15. Tampilan Grafik Integrasi Grafis dan Tahapan Langsung.

Jika pengguna memilih menu tombol *View Graphic* pada Gambar 14, maka akan muncul tampilan seperti Gambar 15.

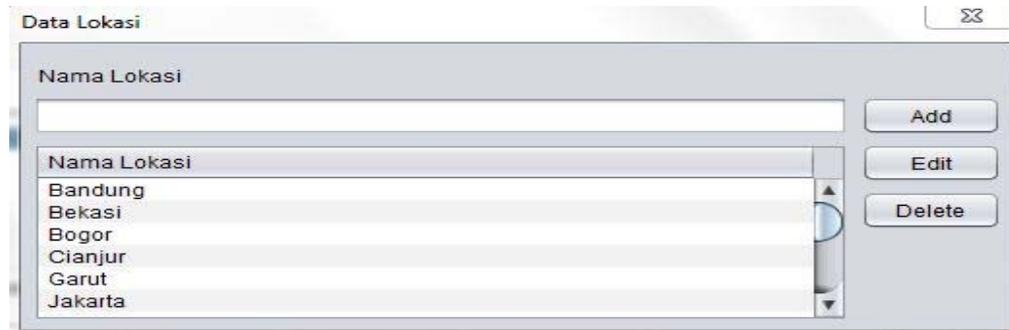
Id Project	Nama Project	Lokasi
1	Tugas Akhir 2	Bogor
2	testing	London
3	adasd	Bekasi
4	asd	Cianjur

Keterangan: Semoga Lulus

Gambar 16. Tampilan *Open Project*.

Jika pengguna memilih menu *Open Project*, maka akan muncul tampilan seperti Gambar 16. Jika pengguna ingin mencari project sesuai lokasi yang dipilih, pengguna bisa memilih didalam *combox* dan menekan tombol OK.

Jika pengguna ingin membuka *project* yang dipilih, klik tombol *Open*. Jika pengguna ingin mengubah nama *project* atau lokasi yang dipilih, klik tombol *Edit*. Jika pengguna ingin menghapus *project* yang dipilih klik tombol *Delete*.



Gambar 17. Tampilan Data Lokasi.

Jika pengguna memilih Data Lokasi, maka akan muncul tampilan seperti Gambar 17. Jika pengguna ingin menambahkan lokasi baru, ketik nama lokasi yang baru kemudian klik tombol *Add*, jika lokasi yang ditambahkan sudah ada maka akan muncul pesan. Jika pengguna ingin mengubah nama lokasi yang ada, klik tombol *Edit*. Jika pengguna ingin menghapus lokasi, pilih lokasi yang mau dihapus kemudian klik tombol *Delete*.

5. PEMBAHASAN DAN UJI COBA PENELITIAN

- Pengguna akan diberi pesan jika salah memasukan data untuk *new project*.
- Pengguna tidak akan salah input data saluran karena sistem akan memberi pesan jika terjadi kesalahan input.
- Pengguna akan diberi pesan saat salah memasukkan data untuk data lokasi.
- Pengguna tidak akan salah memasukan data detail saluran karena sistem akan memberi pesan jika terjadi kesalahan input.

6. SIMPULAN

Berdasarkan hasil penelitian dan pengujian dapat disimpulkan:

1. Aplikasi profil aliran hasil dari penelitian ini dapat digunakan untuk membantu penggunaan, mengolah data-data dan menghitung profil aliran.
2. Hasil penelitian memperlihatkan bawah cara perhitungan antara di kondisi sebenarnya (dilapangan) dan berdasarkan persamaan-persamaan sesuai dasar teori adalah sama.

DAFTAR PUSTAKA

1. Ambler, S. W. (2004). *Agile Model Driven Development with UML 2*. Cambridge: Cambridge University Press.
2. Chonoles, M. (2003). *UML 2 for Dummies*. Indiana: Wiley Publishing, Inc.
3. Chow, Ven Te. (1992). *Hidrolika Saluran Ternuka*. Jakarta: Erlangga.
4. Fathansyah. (2007). *Basis Data*. Bandung: Informatika.
5. Kadir, Abdul. (2003). *Dasar Pemrograman JavaTM 2*. Yogyakarta: Andi Offset.
6. Nugroho, A. (2004). *Konsep Pengembangan Sistem Basis Data*. Bandung: Penerbit Informatika.
7. Triatmodjo, Bambang. (1993). *Hidrolika I*. Yogyakarta: Beta Offset.
8. Triatmodjo, Bambang. (1993). *Hidrolika II*. Yogyakarta: Beta Offset.