

GDK4201

Studi dan Integrasi Workflow menggunakan BPMS dan Sistem Email

Latar Belakang

- Workflow merupakan pemodelan proses bisnis, dapat digambarkan sebagai flow map atau BPMN
- Workflow dapat diotomasi menggunakan BPMS, seperti Camunda.
- Agar eksekusi workflow lebih alamiah dengan model komunikasi organisasi saat ini, maka event-eventnya dapat dipropagasi dan diintegrasikan dengan sistem email.

Masalah yang dibahas

- Bagaimana memodelkan workflow dengan BPMN?
- Event-event workflow apa saja yang dapat dipropagasi ke sistem email?
- Bagaimana mekanisme propagasi dan integrasi workflow dengan sistem email?

Yang harus dilakukan mahasiswa

- Studi tentang proses bisnis, workflow, pemodelan BPMN, BPMS, Sistem Email
- Memodelkan suatu proses bisnis tertentu menggunakan BPMN
- Mengidentifikasi event-event dari workflow yang 'perlu' dipropagasi via sistem email.
- Mengimplementasikan integrasi BPMS dengan Sistem Email.

JNH4203

Pembangkit Papan Permainan Ular Tangga Acak

Tujuan Skripsi:

Membuat generator papan permainan Ular Tangga dengan spesifikasi berikut:

- Output berupa gambar papan. Tangga digambarkan dengan garis lurus. Ular digambarkan dengan kurva yang agak bergelombang.
- Ukuran papan, jumlah ular, dan jumlah tangga ditentukan dari parameter input
- Semua ular dan tangga tidak ada yang bertumpukan atau berpotongan
- Papan yang dihasilkan harus Nampak “bagus” seperti papan yang digambar manual oleh manusia.

Yang harus dikerjakan:

- Menentukan kriteria papan ular tangga yang “bagus”
- Merancang algoritma untuk membangkitkan papan Ular Tangga acak
- Mengimplementasikan algoritma tersebut



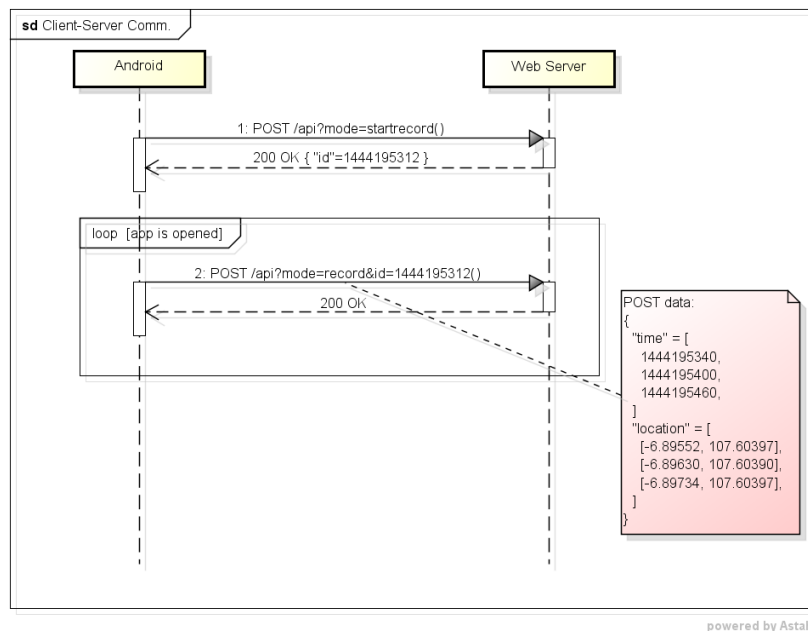
PAN4204]

Perekam Pergerakan Posisi Android, Studi Kasus: KIRI Android App

- ▶ KIRI Smart Public Transport adalah aplikasi navigasi yang tersedia di Google Play [[link](#)], dan kode sumber nya di Github [[link](#)].
- ▶ Saat digunakan, aplikasi secara berkala mendeteksi posisi pengguna.
- ▶ Ada kebutuhan untuk merekam posisi ini dan mencatatnya di server untuk analisa:
 - ▶ Posisi diambil setiap x detik sekali
 - ▶ Data dikirimkan setiap y detik sekali ($y > x$)
 - ▶ Server menerima dan memasukkan ke basis data (lihat di sini)
- ▶ Skripsi menambah fitur ini di kode sumber yang sudah ada, plus membuat penangkapnya di server dengan Codelgniter.

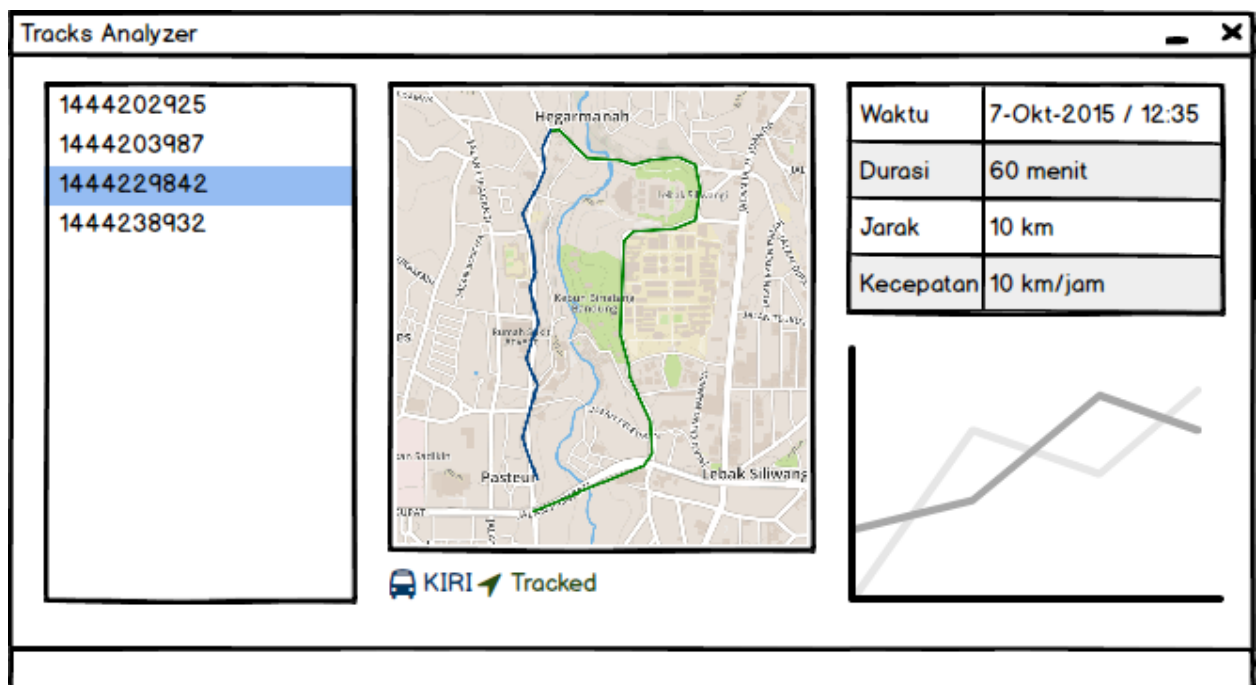
Teknologi

- ▶ Android Development & Android Studio
- ▶ Location Services in Android
- ▶ [Codelgniter](#)
- ▶ HTTP Protocol & Web Services



[PAN4205] Visualisasi dan Analisa Hasil Rekaman Posisi Android dengan JavaFX

- ▶ Topik PAS4002 Mencatat pergerakan pengguna aplikasi KIRI.
- ▶ Topik PAS4003 ini memvisualisasikan data tersebut untuk analisa:
 - ▶ Path hasil GPS
 - ▶ Path angkutan umum asli dari KIRI
 - ▶ Grafik Kecepatan
 - ▶ Data standar: waktu pengambilan, lama pengambilan, jarak pengambilan, kecepatan rata-rata.



- ▶ JavaFX (bisa pakai [SceneBuilder 2.0](#)) (termasuk chart)
- ▶ [GMapsFX](#)
- ▶ MySQL & JDBC

[Tambahan] Format Pencatatan Posisi pada Basis Data

```
CREATE TABLE IF NOT EXISTS `tracking` (  
  `trackingId` int(11) NOT NULL,  
  `timestamp` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP
```

```

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `trackingdetail` (

  `trackingId` int(11) NOT NULL,

  `secondsSinceStart` int(11) NOT NULL,

  `position` point NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tracking`

ALTER TABLE `trackingdetail`

  ADD PRIMARY KEY (`trackingId`,`secondsSinceStart`), ADD KEY `trackingId` (`trackingId`);

ALTER TABLE `trackingdetail`

ADD CONSTRAINT `trackingdetail_tracking` FOREIGN KEY (`trackingId`) REFERENCES `tracking`
(`trackingId`) ON DELETE CASCADE ON UPDATE CASCADE;

```



PENGANTAR ALLOY

- Alloy merupakan perangkat lunak untuk verifikasi.
- Menggunakan *relational logic*.
- Memodelkan asumsi dalam *fact*.
- Memodelkan *constraint* dalam *pred*.
- Memodelkan ekspresi dalam *fun*.
- Memodelkan *property* yang ingin diuji dalam *assert*.

```

sig Platform {}
there are "Platform" things

sig Man {ceiling, floor: Platform}
each Man has a ceiling and a floor Platform

pred Above [m, n: Man] {m.floor = n.ceiling}
Man m is "above" Man n if m's floor is n's ceiling

fact {all m: Man | some n: Man | Above[n,m] }
"One Man's Ceiling Is Another Man's Floor"

assert BelowToo {
  all m: Man | some n: Man | Above [m,n]
}
"One Man's Floor Is Another Man's Ceiling"?

check BelowToo for 2
check "One Man's Floor Is Another Man's Ceiling"
counterexample with 2 or less platforms and men?

```

- clicking "Execute" ran this command
 - counterexample found, shown in graphic

PENGANTAR ALLOY

- Mampu menunjukkan *instance* dari *pred*.

```

sig Platform{}
sig Man{ceiling,floor: Platform}

pred Above[m,n: Man] {m.floor = n.ceiling}

fact {all m: Man | some n: Man | Above[n,m] }

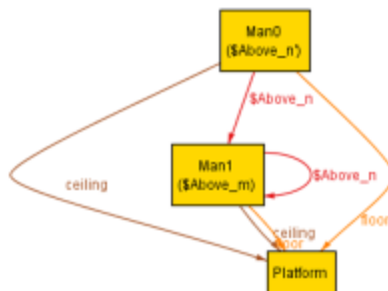
run Above for 2

assert BelowToo{
  all m: Man | some n: Man | Above[m,n]
}

check BelowToo for 2

```

\$Above_n: 2
ceiling: 2
floor: 2



PENGANTAR ALLOY

- Mampu menunjukkan *instance* dari *pred*.
- Mampu menunjukkan *counterexample* dari *assert*.

```

sig Platform{}
sig Man{ceiling,floor: Platform}

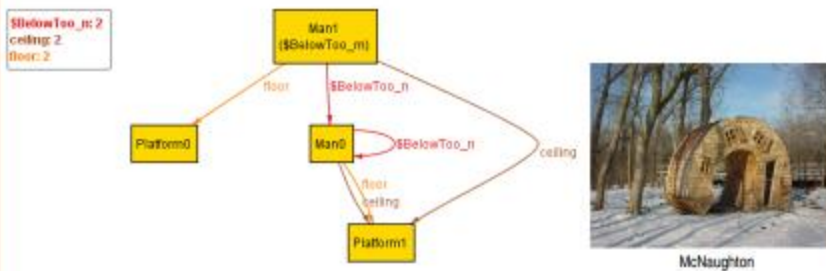
pred Above[m,n: Man] {m.floor = n.ceiling}

fact {all m: Man | some n: Man | Above[n,m] }

run Above for 2

assert BelowToo{
  all m: Man | some n: Man | Above[m,n]
}

check BelowToo for 2
    
```



27

Latar Belakang

Sistem transportasi untuk memindahkan orang/barang antar lantai gedung.

Memodelkan orang/barang, dan elevator dalam concurrency.

Rumusan Masalah:

- Bagaimanakah cara kerja multiple elevators system?
- Bagaimanakah simulasi sistem bekerja?
- Properti apa saja yang menentukan kualitas sistem?
 - Tidak overload
 - Mendahulukan lantai terdekat
 - Melayani semua orang secara adil

Tujuan:

- Menganalisis cara kerja sistem.
- Mengembangkan perangkat lunak simulasi sistem.
- Menguji properti dengan Alloy.