

**ANALISIS DAN PERANCANGAN SISTEM INDEKOS
MENGUNAKAN METODE UNIFIED MODELING LANGUAGE (UML)**

Skripsi

untuk memenuhi sebagian persyaratan

mencapai derajat Sarjana S-1

Program Studi Teknik Informatika



diajukan oleh

Muhammad Fuad Adib

10650008

kepada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UIN SUNAN KALIJAGA

YOGYAKARTA

2014

PENGESAHAN SKRIPSI

Universitas Islam Negeri Sunan Kalijaga

FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR


Nomor : UIN.02/D.ST/PP.01.1/ 267 /2014

Skripsi/Tugas Akhir dengan judul : Analisis dan Perancangan Sistem Indekos Menggunakan Metode Unified Modeling Language (UML)

Yang dipersiapkan dan disusun oleh :
Nama : Muhammad Fuad Adib
NIM : 10650008
Telah dimunaqasyahkan pada : Kamis, 23 Januari 2014
Nilai Munaqasyah : A
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga


TIM MUNAQASYAH :

Ketua Sidang



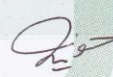
Sumarsono, M.Kom
NIP. 19710209 200501 1 003

Penguji I




Bambang Sugiantoro, M.T
NIP.19751024 200912 1 002

Penguji II



Nurochman, M.Kom
NIP. 19801223 200901 1 007

Yogyakarta, 23 Januari 2014
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan



Akh. Minhaji, M.A, Ph.D
NIP: 19580919 198603 1 002



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal :
Lamp :

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

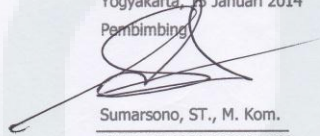
Nama : M. Fuad Adib
NIM : 10650008
Judul Skripsi : Analisis dan Perancangan Sistem Indekos Menggunakan Metode Unified Modeling Language (UML)

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam SARJANA KOMPUTER S.KOM

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 15 Januari 2014
Pembimbing


Sumarsono, ST., M. Kom.
NIP. 19710209 200501 003

PERNYATAAN

iii

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta,



METERAI
TEMPEL
Pajak Penghasilan
CF7B0ACE135604537
6000
DJP
Muhammad Fuad Adib
10650008

KATA PENGANTAR

Assalamu'alaikumWr. Wb.

Segala puji bagi Allah swt yang maha pengasih dan lagi maha penyayang, dengan segala kasih dan sayang-Nya sehingga penyusunan skripsi dengan judul “Analisis dan Perancangan Sistem Indekos Menggunakan Metode *Unified Modeling Language* (UML)” dapat berjalan sebagaimana mestinya. Shalawat dan salam kita tunjukkan kepada guru umat manusia, seorang revolusioner sejari, yaitu Nabi Muhammad SAW yang dalam sejarah kehidupannya, mengajarkan banyak hal kepada kita, salah satunya adalah kemampuan untuk tetap istiqomah berada di jalan-Nya.

Sebagai salah satu syarat memperoleh gelar kesarjanaan pada program studi Teknik Informatika, penyusunan skripsi adalah hal mutlak bagi setiap mahasiswa yang ingin mendapatkan gelar sarjana. Mendapatkan gelar sarjana adalah sesuatu hal yang tidak bisa dilepaskan dari niat awal seseorang kuliah, gelar sarjana juga adalah balasan kita, mahasiswa, terhadap segala bentuk pengorbanan yang diberikan orang tua kita, walaupun sejatinya itu tidak cukup.

Kalaupun harus didedikasikan, maka skripsi ini penulis dedikasikan kepada kedua orang tua yang sangat dicintai, Bapak dan Ibu (Fathullah dan Hamidah), yang tak henti – hentinya memberikan do'a kepada anak – anaknya agar kelak sukses dan berguna, sudah banyak pengorbanan yang mereka lakukan untuk semua anaknya, semoga Allah membalas jasa – jasa mereka Aamiin. Kepada kakak (Kamal) dan adik – adik (Adil, Azha, Wardah dan Wafi) yang

sangat saya sayangi, yang tak henti – hentinya memberikan motivasi. Selanjutnya penulis tidak lupa menghaturkan banyak terima kasih kepada semua pihak yang telah membantu dalam penyusunan baik secara langsung maupun tidak langsung. Sebagai rasa hormat dan ucapan terima kasih penyusun sampaikan kepada:

1. Bapak Prof. Dr. H. Akh. Minhaji, M.A., Ph.D., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta beserta para dosen dan seluruh karyawan/staf pegawai atas bantuan yang diberikan selama penulis mengikuti studi.
2. Bapak Agus Mulyanto, S.Si., M.Kom., selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
3. Bapak Nurochman, S.Kom., M.Kom., selaku Dosen Pembimbing Akademik.
4. Bapak Sumarsono, ST., M.Kom., selaku Dosen Pembimbing yang telah memberikan arahan dan bimbingan selama penelitian.
5. Bapak dan Ibu Dosen TIF UIN SUKA yang telah mendidik dan mengajarkan tentang semua, terima kasih telah bersusah payah mendidik kami dengan sungguh – sungguh.
6. Teman – teman TIF UIN SUKA umumnya dan khususnya INFORMATICS ENGINEERING 2010 (MONSTER INFORMATICS) yang telah mengisi hari – hari selama perkuliahan.

Penulis menyadari masih banyak kekurangan dan kelemahan dalam penelitian ini. Oleh karena itu demi perkembangan penelitian selanjutnya penulis sangat

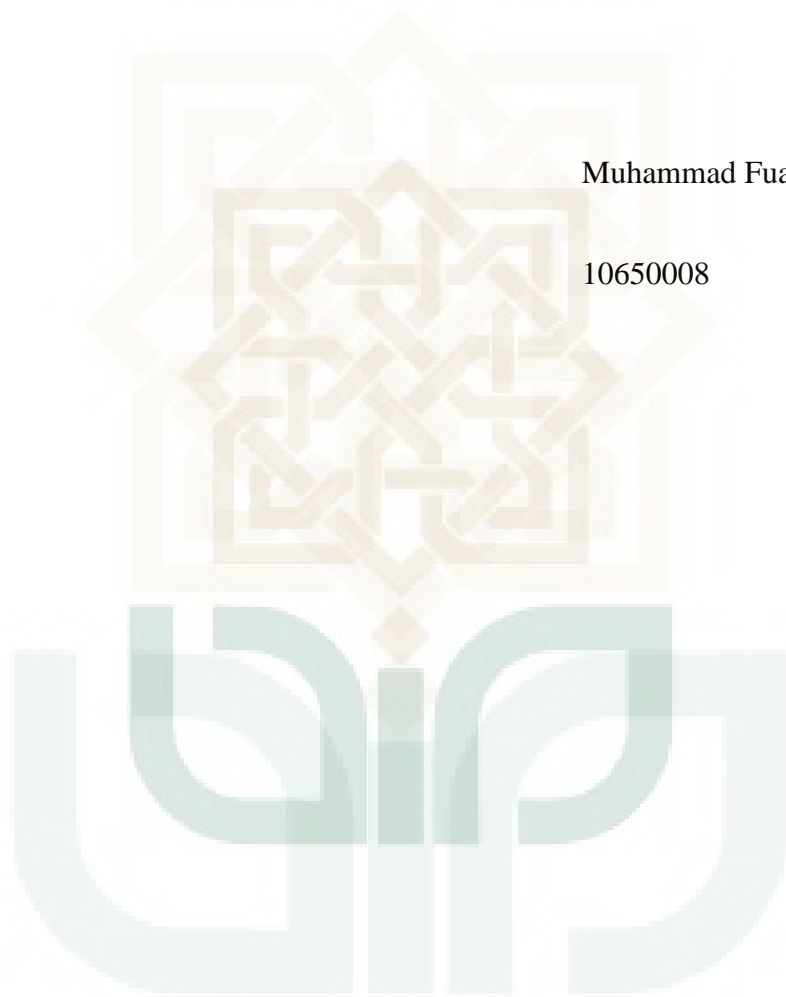
mengharap kritik dan saran dari pembaca. Akhirnya semoga penelitian ini bermanfaat bagi pembaca.

Wassalamu'alaikumWr. Wb

Yogyakarta, 08 Nopember 2013

Muhammad Fuad Adib

10650008



DAFTAR ISI

HALAMAN JUDUL.....	i
PENGESAHAN SKRIPSI	ii
PERNYATAAN.....	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
INTISARI.....	xvi
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	4
C. Batasan Masalah.....	4
D. Tujuan Penelitian	5
E. Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	6
A. Tinjauan Pustaka	6
B. Landasan Teori.....	9

1. Berorientasi Objek.....	9
2. Unified Modeling Language (UML).....	9
3. Diagram <i>Use Case</i>	10
4. Diagram Aktivitas	11
5. Diagram sekuensial	11
6. Diagram Kelas	12
7. Diagram Entity Relationship (E-R).....	13
8. MySQL.....	13
9. Global Positioning System (GPS).....	15
10. Android.....	15
11. Google Maps.....	16
BAB III METODE PENGEMBANGAN SISTEM.....	17
A. Metode Pengembangan Sistem.....	17
1. Pengumpulan Data	17
2. Analisis Sistem.....	17
3. Perancangan dan Pemrograman Sistem.....	18
4. Pengembangan dan Pengujian Sistem.....	18
B. Alat Penelitian.....	19
1. Perangkat Lunak (<i>Software</i>).....	19
2. Perangkat Keras (<i>Hardware</i>).....	19

C. Arsitektur Sistem Indekos	20
BAB IV ANALISIS DAN PENGEMBANGAN SISTEM.....	21
A. Unified Modeling Language (UML).....	21
1. Diagram <i>Use Case</i>	21
2. Diagram Aktivitas	22
3. Diagram Sekuensial.....	37
4. Diagram Kelas	42
B. Perancangan Basis data (<i>Database</i>)	43
1. Diagram Entity Relationship (E-R).....	43
2. Perancangan Tabel	48
C. Perancangan Antarmuka	54
1. Antarmuka Admin	54
2. Antarmuka Pemilik Indekos	58
3. Antarmuka Pengguna (<i>User</i>).....	63
BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM.....	68
A. Implementasi Sistem	68
1. Basis data (<i>Database</i>).....	68
2. Aplikasi Website server.....	75
3. Aplikasi Mobile client	83
B. Pengujian Sistem.....	89

1. Pengujian alpha	89
2. Pengujian beta	90
C. Rencana Pengujian	91
1. Pengujian Alpha	91
2. Pengujian Beta.....	91
BAB VI HASIL DAN PEMBAHASAN	93
A. Hasil Pengujian Sistem	93
B. Pengujian Alpha.....	93
C. Pengujian Beta	94
1. Hasil pengujian pemilik indekos	94
2. Hasil pengujian pengguna indekos.....	94
BAB VII PENUTUP	96
A. Kesimpulan	96
B. Saran.....	96
DAFTAR PUSTAKA	98
LAMPIRAN.....	99

DAFTAR TABEL

1. Tabel 2.1 (Perbandingan studi Pustaka)	8
2. Tabel 2.2 (Daftar simbol diagram <i>use case</i>)	10
3. Tabel 2.3 (Daftar simbol diagram aktivitas)	11
4. Tabel 2.4 (Daftar simbol diagram sekuensial)	12
5. Tabel 2.5 (Daftar simbol diagram kelas)	12
6. Tabel 4.1 (Provinsi)	48
7. Tabel 4.2 (Kab_kota)	48
8. Tabel 4.3 (Pemilik)	49
9. Tabel 4.4 (Lupa_password)	49
10. Tabel 4.5 (Indekos)	50
11. Tabel 4.6 (Kamar_fasilitas_int)	50
12. Tabel 4.7 (Fasilitas_master)	51
13. Tabel 4.8 (Fasilitas_eks)	51
14. Tabel 4.9 (Fasilitas_int)	51
15. Tabel 4.10 (Kamar)	52
16. Tabel 4.11 (Indekos_fasilitas_eks)	53
17. Tabel 4.12 (Admin)	53
18. Tabel 5.1 (Rencana pengujian alpha)	89
19. Tabel 5.2 (Rencana pengujian beta bagian pemilik)	90
20. Tabel 5.3 (Rencana pengujian beta bagian pengguna)	90
21. Tabel 5.4 (Daftar penguji tahap alpha)	91
22. Tabel 5.5 (Daftar penguji tahap beta pemilik indekos)	92
23. Tabel 5.6 (Daftar Penguji tahap beta pengguna indekos)	92
24. Tabel 6.1 (Hasil pengujian sistem tahap alpha)	93
25. Tabel 6.2 (Hasil pengujian sistem pemilik indekos)	94
26. Tabel 6.3 (Hasil pengujian sistem pengguna indekos)	94

DAFTAR GAMBAR

1. Gambar 2.1 (Daftar simbol diagram E-R).....	13
2. Gambar 3.1 (Arsitektur sistem indekos)	18
3. Gambar 4.1 (Diagram <i>Use Case</i> admin)	19
4. Gambar 4.2 (Diagram <i>Use Case</i> pengguna indekos)	19
5. Gambar 4.3 (Diagram <i>Use Case</i> pemilik indekos)	20
6. Gambar 4.4 (Diagram Aktivitas admin provinsi)	21
7. Gambar 4.5 (Diagram Aktivitas admin kab kota)	22
8. Gambar 4.6 (Diagram Aktivitas admin fasilitas master)	23
9. Gambar 4.7 (Diagram Aktivitas admin fasilitas eksternal)	24
10. Gambar 4.8 (Diagram aktivitas pemilik data pribadi/profil)	25
11. Gambar 4.9 (Diagram aktivitas pemilik ubah password)	26
12. Gambar 4.10 (Diagram aktivitas pemilik indekos)	27
13. Gambar 4.11 (Diagram aktivitas pemilik kamar)	28
14. Gambar 4.12 (Diagram aktivitas fasilitas internal)	29
15. Gambar 4.13 (Diagram aktivitas kamar fasilitas internal)	30
16. Gambar 4.14 (Diagram aktivitas pemilik kontrak)	31
17. Gambar 4.15 (Diagram aktivitas indekos fasilitas eksternal)	32
18. Gambar 4.16 (Diagram aktivitas pengguna indekos terdekat)	33
19. Gambar 4.17 (Diagram aktivitas pengguna pencarian indekos kamar)	34
20. Gambar 4.18 (Diagram aktivitas pengguna sinkronisasi data)	35
21. Gambar 4.19 (Diagram aktivitas pengguna rute indekos)	36
22. Gambar 4.20 (Diagram sekuensial admin)	37
23. Gambar 4.21 (Diagram sekuensial pemilik indekos kamar)	38
24. Gambar 4.22 (Diagram sekuensial pemilik kontrak)	39
25. Gambar 4.23 (Diagram sekuensial pengguna cari indekos, rute indekos)	40
26. Gambar 4.24 (Diagram sekuensial pengguna sinkronisasi data)	41
27. Gambar 4.25 (Diagram kelas)	42
28. Gambar 4.26 (Diagram <i>Entity Relationship</i> E-R)	43
29. Gambar 4.27 (Diagram E-R provinsi, kab_kota dan pemilik)	44
30. Gambar 4.28 (Diagram E-R kab_kota, indekos dan pemilik)	44
31. Gambar 4.29 (Diagram E-R indekos, fasilitas_eks dan kamar)	45
32. Gambar 4.30 (Diagram E-R pemilik, fasilitas_int dan kamar)	45
33. Gambar 4.31 (Diagram E-R kab_kota, fasilitas_eks_dan fasilitas_master)	46
34. Gambar 4.32 (Diagram E-R entitas admin)	46
35. Gambar 4.33 (Diagram lain <i>Entity Relationship</i>)	47
36. Gambar 4.34 (Halaman login admin)	54
37. Gambar 4.35 (Halaman depan <i>homepage</i>)	54
38. Gambar 4.36 (Halaman provinsi)	55
39. Gambar 4.37 (Antarmuka halaman kab kota).....	56
40. Gambar 4.38 (Halaman fasilitas master)	56
41. Gambar 4.39 (Halaman fasilitas eksternal)	57

42. Gambar 4.40 (Halaman login, daftar konfirmasi email dan lupa password)	58
43. Gambar 4.41 (Halaman pemilik pengisian data pribadi)	59
44. Gambar 4.42 (Halaman depan <i>homepage</i>)	59
45. Gambar 4.43 (Halaman profil)	60
46. Gambar 4.44 (Halaman ubah password)	60
47. Gambar 4.45 (Halaman depan indekos)	61
48. Gambar 4.46 (Halaman tambah indekos)	61
49. Gambar 4.47 (Halaman kamar)	62
50. Gambar 4.48 (Halaman kontrak)	62
51. Gambar 4.49 (Halaman <i>Splash screen</i>)	63
52. Gambar 4.50 (Halaman depan <i>homepage</i>)	64
53. Gambar 4.51 (Halaman sinkronisasi data)	64
54. Gambar 4.52 (Halaman cari indekos)	65
55. Gambar 4.53 (Halaman daftar indekos)	65
56. Gambar 4.54 (Halaman detail indekos)	66
57. Gambar 4.55 (Halaman rute indekos).....	66
58. Gambar 4.56 (Halaman detail kamar).....	67
59. Gambar 5.1 (<i>Database db_indekos</i>)	68
60. Gambar 5.2 (Struktur tabel admin)	69
61. Gambar 5.3 (Struktur tabel pemilik)	69
62. Gambar 5.4 (Struktur tabel provinsi)	70
63. Gambar 5.5 (Struktur tabel kab_kota)	70
64. Gambar 5.6 (Struktur tabel indekos)	71
65. Gambar 5.7 (Struktur tabel kamar)	71
66. Gambar 5.8 (Struktur tabel fasilitas_master)	72
67. Gambar 5.9 (Struktur tabel fasilitas_eks)	72
68. Gambar 5.10 (Struktur tabel fasilitas_int)	73
69. Gambar 5.11 (Struktur tabel indekos_fasilitas_eks)	73
70. Gambar 5.12 (Struktur tabel kamar_fasilitas_int)	74
71. Gambar 5.13 (Struktur tabel lupa_password)	74
72. Gambar 5.14 (Tampilan admin login)	75
73. Gambar 5.15 (Tampilan admin provinsi).....	75
74. Gambar 5.16 (Tampilan admin kab kota)	76
75. Gambar 5.17 (Tampilan admin fasilitas eksternal).....	76
76. Gambar 5.18 (Tampilan admin fasilitas master)	77
77. Gambar 5.19 (Tampilan pemilik login)	77
78. Gambar 5.20 (Tampilan pengisian data pribadi)	78
79. Gambar 5.21 (Tampilan pemilik lupa password)	78
80. Gambar 5.22 (Tampilan pemilik indekos)	79
81. Gambar 5.23 (Tampilan pemilik indekos detail)	79
82. Gambar 5.24 (Tampilan pemilik fasilitas eksternal)	80
83. Gambar 5.25 (Tampilan pemilik kamar)	80
84. Gambar 5.26 (Tampilan pemilik kamar tambah)	81
85. Gambar 5.27 (Tampilan pemilik kamar fasilitas internal).....	81
86. Gambar 5.28 (Tampilan pemilik fasilitas internal)	82

87. Gambar 5.29 (Tampilan pemilik kontrak)	82
88. Gambar 5.30 (Tampilan pengguna <i>splash screen</i>)	83
89. Gambar 5.31 (Tampilan halaman depan <i>homepage</i>)	83
90. Gambar 5.32 (Tampilan sinkronisasi data)	84
91. Gambar 5.33 (Tampilan daftar provinsi)	84
92. Gambar 5.34 (Tampilan daftar kab kota)	85
93. Gambar 5.35 (Tampilan daftar fasilitas master)	85
94. Gambar 5.36 (Tampilan daftar fasilitas eksternal)	86
95. Gambar 5.37 (Tampilan daftar indekos)	86
96. Gambar 5.38 (Tampilan detail indekos)	87
97. Gambar 5.39 (Tampilan rute indekos)	88
98. Gambar 5.39 (Tampilan detail kamar)	88



DAFTAR LAMPIRAN

A. Daftar kuesioner pemilik indekos	99
B. Daftar sekolah SMAN Daerah Istimewa Yogyakarta	104
C. Daftar Universitas Negeri, Swasta dan Institut di Daerah Istimewa Yogyakarta	106
D. Daftar potongan <i>Source Code</i> Program	107



ANALISIS DAN PERANCANGAN SISTEM INDEKOS

MENGGUNAKAN METODE UNIFIED MODELING LANGUAGE (UML)

Muhammad Fuad Adib
10650008

INTISARI

Indekos adalah sebuah jasa yang menawarkan sebuah kamar/tempat untuk ditinggali dengan sejumlah pembayaran teruntuk setiap periode, bagi pemilik indekos sudah banyak melakukan promosi agar indekosnya dapat dengan mudah dicari oleh pencari indekos, namun selama ini masih kurangnya promosi yang dilakukan seperti penyebaran iklan, website dan lainnya. Dan masih belum adanya yang bisa mengatur setiap kamar yang ada dalam indekos yang dimiliki, dan juga belum adanya yang dapat mengatur waktu sewa – menyewa bagi setiap kamar. Dan bagi pencari indekos mencari indekos disuatu tempat tidaklah mudah apalagi sesuai dengan fasilitas yang diinginkan. Tujuan dari penelitian ini agar pemilik indekos dapat mempromosikan indekosnya, dan dapat mengatur setiap kamar yang ada dan juga dapat mengatur waktu sewa – menyewa bagi setiap kamar, dan tujuan bagi pengguna/pencari indekos agar dapat dengan mudah mencari indekos yang sesuai dengan fasilitas yang diinginkan.

Metode penelitian yang dilakukan adalah dengan wawancara terhadap para pengguna aplikasi, analisis kebutuhan hasil dari wawancara, perancangan dan implementasi program. Perancangan sistem menggunakan UML (*Unified Modeling Language*) dan sistem dijalankan pada dua *platform* yaitu *Website* sebagai *server* dan *Android* untuk pengguna.

Sistem yang dihasilkan bagi pemilik indekos dapat menjadi tempat untuk mempromosikan, mengatur setiap kamar dan mengatur sewa – menyewa kamar. dan bagi pengguna/pencari indekos sistem yang dihasilkan dapat mencari indekos sesuai fasilitas yang diinginkan.

Kata kunci: Indekos, Server, Android, UML

SYSTEM ANALYSIS AND DESIGN LODGER
USING METHOD UNIFIED MODELING LANGUAGE (UML)

Muhammad Fuad Adib
10650008

ABSTRACT

Lodger is a service that offers a room / place to live with a number of payments each period will belong, for for homestay owners have been doing a lot of promotion so that his lodger can be easily searched by search lodger, however, for this is still lacking was the campaign undertaken such as the spread of advertising, website and other. And still the lack of which could regulate every room in homestay owned, and also the absence of which can set the lease time for each room. And the search for a boarder looking for a lodger in one place is not easy especially in accordance with the desired facility. The purpose of this research in order to promote the homestay owner of his lodger, and can organize every room there and also to set the time of the lease for each room. And objectives for the user / searcher lodger, to easily search for lodger in accordance with the desired facility.

Research methodology is the interview of the user application , the results of the needs analysis interview , program design and implementation . System design using UML (Unified Modeling Language) and the system is run on the Android platform.

The resulting system for owners lodger can be a place to promote, organize and arrange each room rental rooms. And for the user / searcher lodger, the resulting system can search for desired facilities boarder appropriate.

Keywords: Lodger , Server , Android , UML

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam Kamus Besar Bahasa Indonesia Indekos adalah sebuah jasa yang menawarkan sebuah kamar/tempat untuk ditinggali dengan sejumlah pembayaran teruntuk setiap priode, bisnis indekos masih sangat menguntungkan, indekos banyak dicari oleh seorang yang ingin tinggal didaerah tertentu yang bukan tempat tinggal aslinya, seperti karyawan yang bekerja diperusahaan tertentu yang jauh jika pulang – pergi dari kantor – kerumah yang akhirnya mencari tempat tinggal yang dekat dengan kantor, atau juga seorang mahasiswa yang kuliah didaerah yang jauh dari tempat tinggalnya, akan mencari tempat tinggal yang dekat dengan kampusnya agar tidak menghabiskan waktu hanya untuk pulang – pergi dan juga agar kuliahnya lebih mudah diatur.

Perusahaan atau instansi seperti kampus, universitas atau sekolah dalam menyeleksi karyawan, mahasiswa atau siswa baru jarang sekali dilihat dari daerah dimana tempat perusahaan atau instansi itu berdiri tapi dari aspek lain, akhirnya mereka yang diterima harus mencari tempat tinggal baru untuk sementara didaerah yang mereka belum tahu sebelumnya, mencari indekos didaerah yang baru sangatlah sulit apalagi belum tentu ketika menemukan indekos, indekos tersebut masih kosong untuk bisa ditinggali.

Mencari indekos tidaklah mudah selain tempatnya yang masih kosong atau tidak, terkadang ada aspek lain seperti fasilitas atau harga, mungkin dapat menemukan indekos yang masih kosong tapi harganya bisa saja mahal dan sangat mahal yang akhirnya batal untuk menempati indekos tersebut, dan mungkin ada juga yang mencari berdasarkan fasilitas yang ada didalam indekos tersebut, seperti televisi, tempat tidur, dan lain sebagainya yang sudah disediakan oleh pemilik indekos. Faktor – faktor tersebut biasanya menjadi salah satu pertimbangan dalam memilih indekos didaerah baru.

Sudah banyak cara promosi – promosi indekos seperti menyebarkan kertas iklan/promosi ditempat – tempat ramai atau membuat pemberitahuan yang dipasang didepan rumahnya, contoh “menerima indekos untuk putra” dan menjadikan website sebagai media promosi untuk indekos dengan keterangan – keterangannya, namun masih ada kelemahan/kekurangan dari cara – cara yang diuraikan diatas, untuk menyebarkan iklan/promosi, perlu seorang kurir untuk yang menyebarkannya dan belum tentu yang mendapatkan iklan/promosi tersebut sedang mencari indekos. Jika memasang iklan/promosi hanya didepan rumah hanya orang – orang yang berada didekat rumah saja yang tahu, dan juga bagi yang mencari belum tahu apa sajakah fasilitas – fasilitas yang ada didalam indekos tersebut. Promosi menggunakan media internet/website lebih baik dari yang sebelumnya informasi bisa dilihat oleh semua orang, namun kebanyakan website masih hanya memberikan informasi pasif, contohnya “info indekos A, terdapat 5 ruangan harga Rp. 100.000/bulan, fasilitas B, C dan D didaerah E”, dari informasi tersebut sudah cukup memadai, namun sebagai pencari indekos tidak

tahu ruangan tersebut berapa yang masih kosong, dan yang sudah ada isinya, dan alamat bagi yang belum tahu daerah baru, akan susah sekali mencari alamat yang diberikan diwebsite, jadi masih terdapat kekurangan dari promosi – promosi tersebut.

Android adalah salah satu *smartphone* yang mendukung GPS (*Global Positioning System*) yang berfungsi sebagai alat koordinat dari suatu tempat atau posisi yang telah diketahui, GPS di *smartphone* sering digunakan untuk menentukan posisi dari tempat berada, dan juga sering digunakan untuk mencari tempat / daerah, atau menghitung jarak dari kota/tempat A ke kota/tempat B berapakah jaraknya. GPS dapat digunakan untuk menentukan letak dimana tempat indekos yang diinginkan, menentukan rute untuk menuju tempat indekos, menghitung jarak terdekat dari pengguna/pencari indekos ke tempat indekos tersebut berada, dan juga menghitung jarak terdekat dari tempat tertentu yang diinginkan, contohnya “ingin mencari indekos yang berada didekat kampus UIN Sunan Kalijaga.”

Implementasi dari program yang akan dibuat berbasis website dan *mobile android* yang berorientasi objek, untuk itu perancangan sistem menggunakan metode *Unified Modeling Language* (UML), karena UML adalah perancangan sistem yang memfokuskan pada pengembangan sistem yang berorientasi objek, menggunakan UML pada pemrograman berorientasi objek akan mempermudah pada saat implementasi program sebab perancangan sistem dan program sistem sama berorientasi objek.

Dari latar belakang tersebut penulis melakukan penelitian untuk merancang dan membuat sistem indekos untuk bisa digunakan bagi seorang pendatang atau perantau dalam mencari indekos yang bisa menjadi tempat tinggalnya sementara di daerah tersebut dengan semua fasilitas yang diinginkan. Dan lebih detail lagi dapat melihat setiap ruangan tersebut masih kosong atau tidak dan melihat setiap fasilitas yang ada didalamnya, dengan rancangan seperti ini diharapkan pencari indekos tidak hanya lihat indekos ada dimana saja tetapi bisa lihat juga masih ada ruangan kosong atau tidak diindekos tersebut dan fasilitas yang ada didalamnya.

B. Rumusan Masalah

1. Bagaimana merancang agar sistem dapat mengatur data ruangan indekos untuk pemilik indekos tersebut
2. Bagaimana merancang sistem untuk mengatur waktu sewa – menyewa ruangan dan atau perpanjangan sewa bagi penyewa/pengguna dan pemilik indekos tersebut
3. Bagaimana menjadikan sistem sebagai tempat mencari indekos yang masih kosong atau masih bisa ditempati

C. Batasan Masalah

1. Sistem untuk *user* dibuat berbasis mobile Android.
2. Sistem untuk pemilik indekos dibuat berbasis website menggunakan *framework PHP Codeigniter*
3. Untuk tampilan website penulis menggunakan *bootstrap*
4. Fasilitas terdapat dua internal dan eksternal, internal adalah fasilitas yang ada didalam kamar, dan eksternal adalah tempat – tempat terdekat indekos, penulis

membatasi kampus/universitas dan sekolah dan membatasi wilayah untuk Sleman, Bantul dan Kota Yogyakarta.

5. Diagram UML yang digunakan adalah diagram *use case*, diagram aktifitas, diagram kelas, diagram sequential dan diagram E-R (Entity Relationship) untuk rancangan basis datanya
6. Untuk keamanan, penulis hanya sampai validasi akun pemilik indekos

D. Tujuan Penelitian

1. Merancang sistem yang dapat mengatur data ruangan indekos untuk pemilik indekos tersebut
2. Merancang sistem yang dapat mengatur waktu sewa – menyewa ruangan dan atau perpanjangan sewa bagi penyewa/pengguna dan pemilik indekos tersebut
3. Merancang sistem agar menjadi tempat untuk mencari indekos yang masih kosong atau masih bisa ditempati.

E. Manfaat Penelitian

Manfaat yang didapatkan dari hasil penelitian ini untuk pencari indekos seperti perantau atau mahasiswa luar kota adalah dapat menjadi tempat untuk mencari indekos di daerah yang akan ditinggali, dengan semua fasilitas yang sesuai keinginan. Dan bagi pemilik indekos dapat menjadi tempat untuk mengiklankan atau mempromosikan indekos agar lebih banyak lagi yang datang untuk menempati, dan juga dapat mengatur setiap ruang bagi pemilik indekos. Dan bagi penulis manfaat dari penelitian ini adalah dapat menjadi ilmu lebih untuk belajar membuat software yang mengikuti aturan siklus hidup pengembangan perangkat lunak (*Software Development Life Cycle*).

BAB VII

PENUTUP

A. Kesimpulan

Berdasarkan hasil dari pengujian yang telah dilakukan penulis pada penelitian mengenai *Analisis dan Perancangan Sistem Indekos Menggunakan Metode Unified Modeling Language (UML)* ini, maka dapat diambil beberapa kesimpulan, yaitu :

1. Sistem dapat mengatur data ruangan indekos untuk pemilik indekos
2. Sistem dapat mengatur waktu sewa – menyewa ruangan dan atau perpanjang sewa bagi penyewa/pengguna dan pemilik indekos
3. Sistem dapat menjadi tempat untuk mencari indekos yang masih kosong atau masih bisa ditempati.

B. Saran

Penelitian yang dilakukan tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, masih perlu pengembangan sistem agar kinerja sistem menjadi lebih baik, antaranya :

1. Dapat menambahkan kategori untuk fasilitas master dan fasilitas eksternal agar lebih banyak pilihan tempat.
2. Membuat akun *members* untuk pengguna/pencari yang memesan/menyewa indekos agar dapat melihat kapan waktu penyewaan akan berakhir dan bisa

melakukan permintaan perpanjangan sewa secara *online* dengan menggunakan akun tersebut.

3. Dapat melakukan peringatan/*alarm* pada mobile pengguna yang sudah memiliki akun pada sistem untuk melakukan/permintaan perpanjangan sewa menyewa.
4. Dapat menambahkan diagram UML pada penelitian selanjutnya



DAFTAR PUSTAKA

- ‘Aini, Syifa Qurrotu. *Sistem Informasi Geografis Berbasis Mobile (Pemetaan Objek Wisata Religi Studi Kasus Jateng-DIY)*. Yogyakarta: Skripsi Jurusan Teknik Informatika UIN Sunan Kalijaga, 2012
- Android. (2013), *What is Android?* Dipetik Desember 31, 2013, dari <http://developer.android.com/about/index.html>
- Arief M, Rudyanto. *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi Offset, 2011
- Arif Huda, Akbarul. *24 Jam!! Pintar Pemrograman Android #1 Ebook Version 2.1*. Yogyakarta: Ebook 2012
- Hanif, Akhmad. *Pencarian Tempat Kos Dengan Teknologi Augmented Reality Berbasis Smartphone Android*. Yogyakarta: Skripsi Jurusan Teknik Informatika UIN Sunan Kalijaga, 2013
- Kadir, Abdul. *Dasar Perancangan dan Implementasi Database Relasional*. Yogyakarta: Andi Offset, 2009
- Kadir, Abdul. *Tuntunan Praktis Belajar Database Menggunakan MySQL*. Yogyakarta: Andi Offset, 2008
- Nugroho, Adi. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Andi Offset, 2009
- Safaat H, Nazruddin. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika, 2011
- Sakur, Stendy B. *PHP5 Pemrograman Berorientasi Objek Konsep & Implementasi*. Yogyakarta Andi Offset, 2010
- Saputra, Hardi. *Implementasi Global Positioning System (GPS) Untuk Pariwisata Daerah Istimewa Yogyakarta Pada Mobile Device Berbasis Android*. Yogyakarta: Skripsi UIN Sunan Kalijaga 2012
- Sholihah. *Sistem Pendukung Keputusan Untuk Pemilihan Pondok Pesantren Sebagai Tempat Tinggal Mahasiswa di D.I. Yogyakarta*. Yogyakarta: Skripsi UIN Sunan Kalijaga, 2012
- Sholihq. *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*. Edisi Pertama Yogyakarta: Graha Ilmu, 2006

LAMPIRAN

A. Daftar kuesioner pemilik indekos

KUESIONER PENGUJIAN SISTEM PEMILIK INDEKOS

Nama : Pardi Hadiwiyono

Alamat : Papingan

No	Item Uji	Penilaian	
		Ya	Tidak
1.	Sistem dapat mengoperasikan data indekos (tambah, ubah dan hapus)	✓	
2.	Sistem dapat mengoperasikan setiap kamar pada indekos (tambah, ubah, dan hapus)	✓	
3.	Sistem dapat melakukan operasi waktu sewa – menyewa dan tambah kontrak, perpanjang dan hapus.	✓	
4.	Sistem dapat menambahkan dan menghapus data fasilitas internal pada tiap kamar	✓	
5.	Sistem dapat menambahkan dan menghapus data fasilitas eksternal pada tiap indekos	✓	
6.	Sistem dapat melakukan operasi pada data fasilitas internal (tambah, ubah dan hapus)	✓	

KUESIONER PENGUJIAN SISTEM PEMILIK INDEKOS

Nama : Ari Winarti

Alamat : Jl. Ori I/8 Papingan .

No	Item Uji	Penilaian	
		Ya	Tidak
1.	Sistem dapat mengoperasikan data indekos (tambah, ubah dan hapus)	✓	
2.	Sistem dapat mengoperasikan setiap kamar pada indekos (tambah, ubah, dan hapus)	✓	
3.	Sistem dapat melakukan operasi waktu sewa – menyewa dan tambah kontrak, perpanjang dan hapus.	✓	
4.	Sistem dapat menambahkan dan menghapus data fasilitas internal pada tiap kamar	✓	
5.	Sistem dapat menambahkan dan menghapus data fasilitas eksternal pada tiap indekos	✓	
6.	Sistem dapat melakukan operasi pada data fasilitas internal (tambah, ubah dan hapus)	✓	

KUESIONER PENGUJIAN SISTEM PEMILIK INDEKOS

Nama: Rudi Harianto

Alamat: Jl. Ori I/16 Papingan

No	Item Uji	Penilaian	
		Ya	Tidak
1.	Sistem dapat mengoperasikan data indekos (tambah, ubah dan hapus)	✓	
2.	Sistem dapat mengoperasikan setiap kamar pada indekos (tambah, ubah, dan hapus)	✓	
3.	Sistem dapat melakukan operasi waktu sewa – menyewa dan tambah kontrak, perpanjang dan hapus.	✓	
4.	Sistem dapat menambahkan dan menghapus data fasilitas internal pada tiap kamar	✓	
5.	Sistem dapat menambahkan dan menghapus data fasilitas eksternal pada tiap indekos	✓	
6.	Sistem dapat melakukan operasi pada data fasilitas internal (tambah, ubah dan hapus)	✓	

KUESIONER PENGUJIAN SISTEM PEMILIK INDEKOS

Nama: Midin

Alamat: Jl. Ori 1/43 papringan

No	Item Uji	Penilaian	
		Ya	Tidak
1.	Sistem dapat mengoperasikan data indekos (tambah, ubah dan hapus)	✓	
2.	Sistem dapat mengoperasikan setiap kamar pada indekos (tambah, ubah, dan hapus)	✓	
3.	Sistem dapat melakukan operasi waktu sewa – menyewa dan tambah kontrak, perpanjang dan hapus.	✓	
4.	Sistem dapat menambahkan dan menghapus data fasilitas internal pada tiap kamar	✓	
5.	Sistem dapat menambahkan dan menghapus data fasilitas eksternal pada tiap indekos	✓	
6.	Sistem dapat melakukan operasi pada data fasilitas internal (tambah, ubah dan hapus)	✓	

KUESIONER PENGUJIAN SISTEM PEMILIK INDEKOS

Nama : Wongso

Alamat : Jl. Sri 1/23 Papingan

No	Item Uji	Penilaian	
		Ya	Tidak
1.	Sistem dapat mengoperasikan data indekos (tambah, ubah dan hapus)	✓	
2.	Sistem dapat mengoperasikan setiap kamar pada indekos (tambah, ubah, dan hapus)	✓	
3.	Sistem dapat melakukan operasi waktu sewa – menyewa dan tambah kontrak, perpanjang dan hapus.	✓	
4.	Sistem dapat menambahkan dan menghapus data fasilitas internal pada tiap kamar	✓	
5.	Sistem dapat menambahkan dan menghapus data fasilitas eksternal pada tiap indekos	✓	
6.	Sistem dapat melakukan operasi pada data fasilitas internal (tambah, ubah dan hapus)	✓	

B. Daftar sekolah SMAN Daerah Istimewa Yogyakarta

No	Nama	Kab Kota
1	SMA Negeri 1	Kotamadya Yogyakarta
2	SMA Negeri 2	Kotamadya Yogyakarta
3	SMA Negeri 3	Kotamadya Yogyakarta
4	SMA Negeri 4	Kotamadya Yogyakarta
5	SMA Negeri 5	Kotamadya Yogyakarta
6	SMA Negeri 6	Kotamadya Yogyakarta
7	SMA Negeri 7	Kotamadya Yogyakarta
8	SMA Negeri 8	Kotamadya Yogyakarta
9	SMA Negeri 9	Kotamadya Yogyakarta
10	SMA Negeri 10	Kotamadya Yogyakarta
11	SMA Negeri 11	Kotamadya Yogyakarta
12	SMA N 1 Cangkringan	Kabupaten Sleman
13	SMA N 1 Depok	Kabupaten Sleman
14	SMA N 1 Gamping	Kabupaten Sleman
15	SMA N 1 Godean	Kabupaten Sleman
16	SMA N Kalasan	Kabupaten Sleman
17	SMA N 1 Minggir	Kabupaten Sleman
18	SMA N 1 Mlati	Kabupaten Sleman
19	SMA N 1 Ngaglik	Kabupaten Sleman
20	SMA N 2 Ngaglik	Kabupaten Sleman
21	SMA N 1 Ngeplak	Kabupaten Sleman
22	SMA N 1 Pakem	Kabupaten Sleman
23	SMA N 1 Prambanan	Kabupaten Sleman
24	SMA N 1 Sayegan	Kabupaten Sleman
25	SMA N 1 Sleman	Kabupaten Sleman
26	SMA N 2 Sleman	Kabupaten Sleman
27	SMA N 1 Tempel	Kabupaten Sleman
28	SMA N 1 Turi	Kabupaten Sleman
29	SMA Negeri 1 Bantul	Kabupaten Bantul
31	SMA Negeri 2 Bantul	Kabupaten Bantul
32	SMA Negeri 3 Bantul	Kabupaten Bantul
33	SMA Negeri 1 Jetis	Kabupaten Bantul
34	SMA Negeri 1 Sewon	Kabupaten Bantul
35	SMA Negeri 1 Banguntapan	Kabupaten Bantul
36	SMA Negeri 2 Banguntapan	Kabupaten Bantul

37	SMA Negeri 1 Bambanglipuro	Kabupaten Bantul
38	SMA Negeri 1 Imogiri	Kabupaten Bantul
39	SMA Negeri 1 Kasihan	Kabupaten Bantul
40	SMA Negeri 1 Kretek	Kabupaten Bantul
41	SMA Negeri 1 Pajangan	Kabupaten Bantul
42	SMA Negeri 1 Piyungan	Kabupaten Bantul
43	SMA Negeri 1 Pleret	Kabupaten Bantul
44	SMA Negeri 1 Pundong	Kabupaten Bantul
45	SMA Negeri 1 Sanden	Kabupaten Bantul
46	SMA Negeri 1 Sedayu	Kabupaten Bantul
47	SMA Negeri 1 Srandakan	Kabupaten Bantul
48	SMA Negeri 1 Dlingo	Kabupaten Bantul

Sumber

[http://id.wikipedia.org/wiki/Daftar sekolah menengah atas di Yogyakarta](http://id.wikipedia.org/wiki/Daftar_sekolah_menengah_atas_di_Yogyakarta)

C. Daftar Universitas Negeri, Swasta dan Institut di Daerah Istimewa Yogyakarta

No	Nama	Kab Kota	Kampus
1	UIN Sunan Kalijaga	Yogyakarta	Negeri
2	Universitas Gadjah Mada	Yogyakarta	Negeri
3	Universitas Negeri Yogyakarta	Yogyakarta	Negeri
4	Universitas Ahmad Dahlan	Yogyakarta	Swasta
5	Universitas Atma Jaya	Yogyakarta	Swasta
6	Universitas Cokroaminoto	Yogyakarta	Swasta
7	Universitas Dirgantara Indonesia	Yogyakarta	Swasta
8	Universitas Islam Indonesia	Yogyakarta	Swasta
9	Universitas Janabadra	Yogyakarta	Swasta
10	Universitas Kristen Duta Wacana	Yogyakarta	Swasta
11	Universitas Kristen Immanuel	Yogyakarta	Swasta
12	Universitas Muhammadiyah Yogyakarta	Yogyakarta	Swasta
13	Universitas Pembangunan Nasional Veteran	Yogyakarta	Swasta
14	Universitas PGRI Yogyakarta	Yogyakarta	Swasta
15	Universitas Proklamasi '45	Yogyakarta	Swasta
16	Universitas Sanata Dharma	Yogyakarta	Swasta
17	Universitas Sarjanawiyata Tamansiswa	Yogyakarta	Swasta
18	Universitas Teknologi Yogyakarta	Yogyakarta	Swasta
19	Universitas Wangsa Manggala	Yogyakarta	Swasta
20	Universitas Widya Mataram	Yogyakarta	Swasta
21	IKIP PGRI Yogyakarta	Yogyakarta	Swasta
22	Institut Pertanian Intan	Yogyakarta	Institut
23	Institut Pertanian Stiper	Sleman	Institut
24	Institut Sains dan Teknologi Akprind	Yogyakarta	Institut
25	Institut Seni Indonesia	Yogyakarta	Institut

Sumber

<http://mfatwah.wordpress.com/2011/04/17/daftar-perguruan-tinggi-negeri-dan-swasta-di-yogyakarta/>

D. Daftar potongan *Source Code* Program

```

NAMA BERKAS = database.php
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = "";
$db['default']['database'] = 'db_indekos';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = "";
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;

/* End of file database.php */
/* Location: ./application/config/database.php */

/=====
NAMA BERKAS = admin.php
<?php

class Admin extends CI_Controller{

    public function cek_data(){
        if($this->session->userdata('admin_login')){
            $this->load->model('admin_m');
            $admin = $this->admin_m->cek_data($this->session-
>userdata('email'));
            if($admin->num_rows()==1){
                $data['admin'] = $admin->result();
                //$this->load->view('admin/data',$data);
                return true;
            }else{
                redirect('admin/login');
            }
        }else{

```

```

        redirect('admin/login');
    }
}

public function provinsi(){
    if($this->cek_data()){
        $this->load->model('daerah_m');
        $this->load->library('pagination');
        $prov = count($this->daerah_m->provinsi());
        $config['base_url'] = base_url().'admin/provinsi';
        $config['total_rows'] = $prov;
        $config['per_page'] = 20;

        $this->pagination->initialize($config);
        $data['page'] = $this->pagination->create_links();
        $data['provinsi'] = $this->daerah_m-
>provinsi_limit_offset($config['per_page'],$this->uri->segment(3));
        $this->load->view('admin/provinsi',$data);
    }
}

public function provinsi_tambah(){
    if($this->cek_data()){
        $this->load->model('daerah_m');
        $this->load->library('form_validation');
        $this->form_validation-
>set_rules('provinsi_kode','Kode','required|trim|numeric');
        $this->form_validation-
>set_rules('provinsi_nama','Nama','required|trim');
        if($this->form_validation->run()){
            if($this->daerah_m->provinsi_tambah()){
                echo"<script
language=\"javascript\">alert('Data berhasil
ditambah.');

```

```

        if($this->cek_data()){
            $provinsi_id = $this->uri->segment(3);
            $this->load->model('daerah_m');
            if($this->daerah_m->provinsi_hapus($provinsi_id)){
                echo"<script    language=\"javascript\">alert('Data
berhasil dihapus.');

```



```

cos(($lat*pi()/180)) *
cos(((indekos_long - $lng)* pi()/180))
)
)*180/pi()
)*60*1.1515
) as jarak
FROM indekos ORDER BY jarak ASC");
$send = "";
while($data = mysql_fetch_array($jarak)){
    $dt = stripslashes($data['jarak']);
    $send .= $dt."#";
}
echo $send;
}
?>
/-----/
NAMA BERKAS = LocalDatabase.java
package com.clientindekos;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class LocalDatabase {
    /* ==== PEMILIK ==== */
    public static final String KEY_ROWID = "_id";
    /* ==== PROVINSI ==== */
    public static final String KEY_PROVINSI_ID = "provinsi_id";
    public static final String KEY_PROVINSI_NAMA = "provinsi_nama";
    public static final String KEY_PROVINSI_KODE = "provinsi_kode";
    public static final String KEY_JARAK = "jarak";
    public static final String KEY_JUMLAH = "jumlah";

    private static final String TAG = "LocalDatabase";
    private DatabaseHelper mDbHelper;
    private SQLiteDatabase mDb;

    private static final String DATABASE_NAMA = "db_indekos";
    private static final String TABEL_PROVINSI = "provinsi";
    private static final int DATABASE_VERSI = 2;

```

```

private final Context mContext;

private static class DatabaseHelper extends SQLiteOpenHelper {

    DatabaseHelper(Context context) {
        // TODO Auto-generated constructor stub
        super(context, DATABASE_NAMA, null,
DATABASE_VERSI);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // TODO Auto-generated method stub
        db.execSQL("CREATE TABLE provinsi(_id integer
primary key autoincrement, provinsi_id integer,provinsi_kode integer,
provinsi_nama text);");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        // TODO Auto-generated method stub
        Log.w(TAG, "Upgrading database dari " + oldVersion + "
ke "
+ newVersion
+ ", upgrading akan menghapus semua data
yang ada.");
        db.execSQL("Drop table if exists provinsi");
        onCreate(db);
    }
}

public LocalDatabase(Context ctx) {
    this.mContext = ctx;
}

public LocalDatabase open() throws SQLException {

    mDbHelper = new DatabaseHelper(mContext);
    mDb = mDbHelper.getWritableDatabase();
    return this;
}

public void close() {

```



```

        mDbHelper.close();
    }

    /* ==== AMBIL DATA UNTUK SINKRONISASI ==== */
    public Cursor select_all_provinsi() {
        return mDb.query(TABEL_PROVINSI, new String[] {
KEY_ROWID,
                                KEY_PROVINSI_ID,  KEY_PROVINSI_KODE,
KEY_PROVINSI_NAMA }, null,
                                null, null, null, KEY_PROVINSI_NAMA);
    }
    // Update sinkronisasi data.
    public boolean updateProvinsi(String provinsi_id, String provinsi_kode,
        String provinsi_nama) {
        ContentValues args = new ContentValues();
        args.put(KEY_PROVINSI_ID, provinsi_id);
        args.put(KEY_PROVINSI_KODE, provinsi_kode);
        args.put(KEY_PROVINSI_NAMA, provinsi_nama);
        return mDb.update(TABEL_PROVINSI, args,
KEY_PROVINSI_ID + "="
                                + provinsi_id, null) > 0;
    }
    // Insert sinkronisasi data.
    public long insertProvinsi(String provinsi_id, String provinsi_kode,
        String provinsi_nama) {
        ContentValues args = new ContentValues();
        args.put(KEY_ROWID, provinsi_id);
        args.put(KEY_PROVINSI_ID, provinsi_id);
        args.put(KEY_PROVINSI_KODE, provinsi_kode);
        args.put(KEY_PROVINSI_NAMA, provinsi_nama);
        return mDb.insert(TABEL_PROVINSI, null, args);
    }
    public boolean deleteProvinsi(String provinsi_id) {
        // TODO Auto-generated method stub
        return mDb.delete(TABEL_PROVINSI, KEY_PROVINSI_ID +
"=" + provinsi_id,
                                null) > 0;
    }
    public Cursor select_provinsi(String provinsi_id) {
        // TODO Auto-generated method stub
        return mDb.query(TABEL_PROVINSI, new String[] {
KEY_ROWID,
                                KEY_PROVINSI_ID,  KEY_PROVINSI_KODE,
KEY_PROVINSI_NAMA },
                                KEY_PROVINSI_ID + "=" + provinsi_id, null,
                                null, null,

```

```

        KEY_PROVINSI_NAMA);
    }
}
/-----/
NAMA BERKAS = SinkronisasiActivity.java
package com.clientindekos;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpRequestBase;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Window;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;

public class SinkronisasiActivity extends Activity {

    private Button mulai_sinkron, kembali;
    private LocalDatabase mDbHelper;
    ProsesSink proses;
    public Handler mHandler;
    //private          String          URL          =
"http://www.android.daarelqurro.sch.id/android.php";
    private String URL = "http://10.0.2.2/indekosclient/android.php";
    public void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder()
                .permitAll().build();
        StrictMode.setThreadPolicy(policy);
        mDbHelper = new LocalDatabase(this);

        mDbHelper.open();
        setContentView(R.layout.sinkron);
        mulai_sinkron = (Button) findViewById(R.id.sinkron_mulai);
        mulai_sinkron.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                // TODO Auto-generated method stub
                proses = new ProsesSink();
                proses.execute();
            }
        });
        kembali = (Button) findViewById(R.id.sinkron_kembali);
        kembali.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }

    class ProsesSink extends AsyncTask<Void, Integer, Void> {

        Dialog dialog;
        ProgressBar progressBar;
        TextView tvLoading, tvPer;
        Button btnCancel;

        protected void onPreExecute() {
            super.onPreExecute();
            dialog = new Dialog(SinkronisasiActivity.this);
            dialog.setCancelable(false);

            dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
            dialog setContentView(R.layout.proses_sink);

            progressBar = (ProgressBar)
dialog.findViewById(R.id.progressBar1);
            tvLoading = (TextView) dialog.findViewById(R.id.tv1);

```

```

tvPer = (TextView) dialog.findViewById(R.id.tvper);
btnCancel = (Button) dialog.findViewById(R.id.btncancel);

btnCancel.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        proses.cancel(true);
        dialog.dismiss();
    }
});

dialog.show();
}

@Override
protected Void doInBackground(Void... params) {
    // TODO Auto-generated method stub
    String[] serverprovinsi_id
        = fetch(URL+"?act=provinsi_id");
    String[] serverprovinsi_kode
        = fetch(URL+"?act=provinsi_kode");
    String[] serverprovinsi_nama
        = fetch(URL+"?act=provinsi_nama");
    String[] clientprovinsi_id = selectProvinsi(1);
    if(serverprovinsi_id.length >= clientprovinsi_id.length){
        for (int i = 0; i < serverprovinsi_id.length; i++) {
            Cursor provinsi =
mDbHelper.select_provinsi(serverprovinsi_id[i]);
            if(provinsi.getCount()==0){
                mDbHelper.insertProvinsi(serverprovinsi_id[i],
serverprovinsi_kode[i], serverprovinsi_nama[i]);
            }else if(provinsi.getCount()==1){
                mDbHelper.updateProvinsi(serverprovinsi_id[i],
serverprovinsi_kode[i], serverprovinsi_nama[i]);
            }else{
                mDbHelper.deleteProvinsi(serverprovinsi_id[i]);
                mDbHelper.insertProvinsi(serverprovinsi_id[i],
serverprovinsi_kode[i], serverprovinsi_nama[i]);
            }
        }
    }
}

```

```

    }
} else {
    for(int i=0;i<clientprovinsi_id.length;i++){

        ArrayList<NameValuePair>postParameters=new
ArrayList<NameValuePair>();
        postParameters.add(new
BasicNameValuePair("provinsi_id",clientprovinsi_id[i]));
        String res=null;
        try{

res=CustomHttpClient.executeHttpPost(URL+"?act=provinsi",postParame
ters);

            String rs=res.toString();
            rs=rs.trim();
            rs=rs.replaceAll("\\s+","");
            if(rs.equals("0")){

mDbHelper.deleteProvinsi(clientprovinsi_id[i]);
            } else {

mDbHelper.updateProvinsi(serverprovinsi_id[i],
serverprovinsi_kode[i], serverprovinsi_nama[i]);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
return null;
}

@Override
protected void onPostExecute(Void result) {
    super.onPostExecute(result);

    dialog.dismiss();

    AlertDialog alert = new AlertDialog.Builder(
        SinkronisasiActivity.this).create();

    alert.setTitle("Sinkronisasi selesai");
    alert.setMessage("Sinkronisasi data berhasil dilakukan");
    alert.setButton("Selesai", new
DialogInterface.OnClickListener() {

```

```

which) {
    public void onClick(DialogInterface dialog, int
        dialog.dismiss();
    }
    });
    alert.show();
}
}

public String LongData(String Data) {
    String LongData = "";
    for (int i = 0; i < Data.length(); i++) {
        if (Data.charAt(i) == ' ') {
            LongData += '~';
        } else {
            LongData += Data.charAt(i);
        }
    }
    return LongData;
}

public String[] selectProvinsi(int column) {
    Cursor provinsi = mDbHelper.select_all_provinsi();
    String result[] = new String[provinsi.getCount()];
    provinsi.moveToFirst();
    int i = 0;
    while (provinsi.isAfterLast() == false) {
        result[i++] = provinsi.getString(column);
        provinsi.moveToNext();
    }
    provinsi.close();
    return result;
}

public String[] fetch(String url) {
    HttpClient httpclient = new DefaultHttpClient();
    HttpRequestBase httpRequest = null;
    HttpResponse httpResponse = null;
    InputStream inputStream = null;
    String response = "";
    StringBuffer buffer = new StringBuffer();
    httpRequest = new HttpGet(url);
    try {
        httpResponse = httpclient.execute(httpRequest);

```

```

    } catch (ClientProtocolException el) {
        el.printStackTrace();
    } catch (IOException el) {
        // TODO Auto-generated catch block
        el.printStackTrace();
    }
    try {
        inputStream = httpResponse.getEntity().getContent();
    } catch (IllegalStateException el) {
        el.printStackTrace();
    } catch (IOException el) {
        el.printStackTrace();
    }
}

byte[] data = new byte[512];
int len = 0;
try {
    while (-1 != (len = inputStream.read(data))) {
        buffer.append(new String(data, 0, len));
    }
} catch (IOException el) {
    el.printStackTrace();
}
try {
    inputStream.close();
} catch (IOException el) {
    el.printStackTrace();
}
response = buffer.toString();
StringParser parser = new StringParser();
ArrayList<Object> output = parser.Parse(response);
Object[] Output = output.toArray();
String[] content = new String[Output.length];
for (int i = 0; i < content.length; i++) {
    content[i] = Output[i].toString();
}
return content;
}
}

/=====
NAMA BERKAS = IndekosTerdekat.java
package com.clientindekos;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;

```

```

import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpRequestBase;
import org.apache.http.impl.client.DefaultHttpClient;

import android.app.ListActivity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.Toast;
import android.widget.AdapterView.AdapterContextMenuInfo;

public class IndekosTerdekat extends ListActivity implements LocationListener {

    private static final int DETAIL = Menu.FIRST;
    private static final int RUTE = Menu.FIRST + 1;
    private static final int BATAL = Menu.FIRST + 2;

    private double terdekatlong, terdekatlat;
    private LocationManager locMgr;
    private LocalDatabase mDbHelper;
    private JarakTerdekat jarakTerdekat;
    //private String IndekosURL =
"http://www.android.daarelqurro.sch.id/android.php";
    private String IndekosURL = "http://10.0.2.2/indekosclient/android.php";

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.cari);

```



```

        mDbHelper = new LocalDatabase(this);
        mDbHelper.open();
        locMgr = (LocationManager)
getSystemService(LOCATION_SERVICE);
        Location loc = locMgr

        .getLastKnownLocation(LocationManager.GPS_PROVIDER);
        terdekat(loc);
        daftarTerdekat();
    }

    public void onResume() {
        super.onResume();

        locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER,
2000, 1,
            this);
    }

    public void onPause() {
        super.onPause();
        locMgr.removeUpdates(this);
    }

    public void onLocationChanged(Location loc) {
        //terdekat(loc);
    }

    public void onProviderDisabled(String provider) {
        Toast.makeText(getApplicationContext(),
            "GPS Tidak aktif, tidak bisa menentukan jarak
terdekat.",
            Toast.LENGTH_SHORT).show();
        finish();
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
    }

    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    private void terdekat(Location loc) {
        if(loc != null){

```

```

        registerForContextMenu(getListView());
        terdekatlat = loc.getLatitude();
        terdekatlong = loc.getLongitude();
        jarakTerdekat = new JarakTerdekat();
        jarakTerdekat.execute();
        //daftarTerdekat();
    }
}

private void daftarTerdekat() {
    Cursor jrk = mDbHelper.get_jarak_terdekat();
    startManagingCursor(jrk);
    String[] from = new String[] {
LocalDatabase.KEY_INDEKOS_NAMA,
        LocalDatabase.KEY_JARAK,
        LocalDatabase.KEY_INDEKOS_UNTUK };
    int[] to = new int[] { R.id.tvlistindekosnama,
        R.id.tvlistindekosuntuk };

    SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,
        R.layout.tvlistindekos, jrk, from, to);
    setListAdapter(adapter);
}

private class JarakTerdekat extends AsyncTask<Void, Integer, Void> {

    private ProgressDialog Dialog;

    protected void onPreExecute() {
        Dialog = new ProgressDialog(IndekosTerdekat.this);
        Dialog.setMessage("Loading terdekat");
        Dialog.show();
    }

    @Override
    protected Void doInBackground(Void... params) {
        // TODO Auto-generated method stub
        String[] serverterdekat_indekos_id = fetch(IndekosURL
            + "?act=terdekat_indekos_id&lat="
terdekatlat + "&lng="
            + terdekatlong);
        String[] serverterdekat_jarak = fetch(IndekosURL
            + "?act=terdekat_jarak&lat=" + terdekatlat +
"&lng="
            + terdekatlong);
    }
}

```

```

String[] clientterdekat = selectHasilTerdekat(1);
Log.d("LATLNG", terdekatlat + " = " + terdekatlong);
if (clientterdekat.length != 0) {
    for (int i = 0; i < clientterdekat.length; i++) {
        Log.d("Hapus", ">>>>>");

        mDbHelper.deleteJarakTerdekat(clientterdekat[i]);
    }
    for (int j = 0; j < serverterdekat_indekos_id.length;
j++) {
        Log.d("LATLNG", terdekatlat + " = " +
terdekatlong);
        Log.d("Insert", "indekos id ="
+
serverterdekat_indekos_id[j] + ", jarak ="
+ serverterdekat_jarak[j]);
        mDbHelper.insertJarakTerdekat(serverterdekat_indekos_id[j],
serverterdekat_jarak[j]);
    }
} else {
    for (int j = 0; j < serverterdekat_indekos_id.length;
j++) {
        Log.d("LATLNG", terdekatlat + " = " +
terdekatlong);
        Log.d("Insert", "indekos id ="
+
serverterdekat_indekos_id[j] + ", jarak ="
+ serverterdekat_jarak[j]);
        mDbHelper.insertJarakTerdekat(serverterdekat_indekos_id[j],
serverterdekat_jarak[j]);
    }
}
return null;
}

protected void onPostExecute(Void result) {
    super.onPostExecute(result);
    Dialog.dismiss();
    AlertDialog alert = new
AlertDialog.Builder(IndekosTerdekat.this)
.create();

    alert.setTitle("Jarak Terdekat");

```

```

        alert.setMessage("Daftar jarak terdekat");
        alert.setButton("Lihat",
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int
which) {
        // TODO Auto-generated method stub
        Dialog.dismiss();
        setContentView(R.layout.cari);
        daftarTerdekat();
    }
});
    alert.show();
}
}

public String[] selectHasilTerdekat(int column) {
    Cursor terdekat = mDbHelper.select_all_jarak_terdekat();
    String result[] = new String[terdekat.getCount()];
    terdekat.moveToFirst();
    int i = 0;
    while (terdekat.isAfterLast() == false) {
        result[i++] = terdekat.getString(column);
        terdekat.moveToNext();
    }
    terdekat.close();
    return result;
}

public String[] fetch(String url) {
    HttpClient httpclient = new DefaultHttpClient();
    HttpRequestBase httpRequest = null;
    HttpResponse httpResponse = null;
    InputStream inputStream = null;
    String response = "";
    StringBuffer buffer = new StringBuffer();
    httpRequest = new HttpGet(url);
    try {
        httpResponse = httpclient.execute(httpRequest);
    } catch (ClientProtocolException el) {
        el.printStackTrace();
    } catch (IOException el) {
        // TODO Auto-generated catch block
        el.printStackTrace();
    }
}

```

```

try {
    inputStream = httpResponse.getEntity().getContent();
} catch (IllegalStateException el) {
    el.printStackTrace();
} catch (IOException el) {
    el.printStackTrace();
}

byte[] data = new byte[512];
int len = 0;
try {
    while (-1 != (len = inputStream.read(data))) {
        buffer.append(new String(data, 0, len));
    }
} catch (IOException el) {
    el.printStackTrace();
}
try {
    inputStream.close();
} catch (IOException el) {
    el.printStackTrace();
}
response = buffer.toString();
StringParser parser = new StringParser();
ArrayList<Object> output = parser.Parse(response);
Object[] Output = output.toArray();
String[] content = new String[Output.length];
for (int i = 0; i < content.length; i++) {
    content[i] = Output[i].toString();
}
return content;
}

public boolean onContextItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case DETAIL:
            AdapterContextMenuInfo detailindekos =
(AdapterContextMenuInfo) item
                .getMenuInfo();
            Intent i = new Intent(this, DetailIndekos.class);
            i.putExtra(LocalDatabase.KEY_INDEKOS_ID,
detailindekos.id);
            startActivity(i);
            return true;
        case RUTE:

```

```

        AdapterContextMenuInfo         ruteindekos         =
(AdapterContextMenuInfo) item
        .getMenuInfo();
        i = new Intent(this, RuteIndekos.class);
        i.putExtra(LocalDatabase.KEY_INDEKOS_ID,
ruteindekos.id);
        startActivity(i);
        return true;
    case BATAL:
        return true;
    }
    return super.onContextItemSelected(item);
}

protected void onItemClick(ListView l, View v, int position, long id) {
    super.onItemClick(l, v, position, id);
    try {
        Log.d("ID INDEKOS", String.valueOf(id).toString());
        Intent i = new Intent(this, DetailIndekos.class);
        i.putExtra(LocalDatabase.KEY_INDEKOS_ID, id);
        startActivity(i);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.terdekat_kembali, menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case R.id.menu_terdekat:
        // Location loc;
        locMgr = (LocationManager)
getSystemService(LOCATION_SERVICE);
        Location loc = locMgr

        .getLastKnownLocation(LocationManager.GPS_PROVIDER);
        terdekat(loc);
        break;
    case R.id.menu_kembali:
        finish();
        break;
    }
}

```

```

    }
    return false;
}

}

/=====/
NAMA BERKAS = RuteIndekos.java
package com.clientindekos;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.webkit.WebView;
import android.widget.Toast;

public class RuteIndekos extends Activity implements LocationListener {

    private Long indekos_id;
    LocalDatabase mDbHelper;
    private LocationManager locMgr;
    private String lt,lg, lt1,lg1;
    private WebView webrute;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.ruteindekos);
        mDbHelper = new LocalDatabase(this);
        mDbHelper.open();
        locMgr
(LocationManager) getSystemService(LOCATION_SERVICE);
        Location loc
locMgr.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        indekos_id = savedInstanceState != null ? savedInstanceState
.getLong(LocalDatabase.KEY_INDEKOS_ID)
:
null;

        if (indekos_id == null) {
            Bundle extras = getIntent().getExtras();
            indekos_id = extras != null ? extras
.getLong(LocalDatabase.KEY_INDEKOS_ID) : null;

```

```

    }
    ruteIndekos(loc);
}

private void detailIndekos() {
    // TODO Auto-generated method stub
    if (indekos_id != null) {
        Cursor cursor = mDbHelper.get_detail_indekos(indekos_id);
        startManagingCursor(cursor);
        lg = cursor.getString(cursor.getColumnIndexOrThrow(LocalDatabase.KEY_INDEKOS_LONG)).toString();
        lt = cursor.getString(cursor.getColumnIndexOrThrow(LocalDatabase.KEY_INDEKOS_LAT)).toString();
    }
}

public void onResume(){
    super.onResume();

    locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER,
    2000, 1, this);
}

public void onPause(){
    super.onPause();
    locMgr.removeUpdates(this);
}

public void onLocationChanged(Location loc){
    ruteIndekos(loc);
}

public void onProviderDisabled(String provider){
    Toast.makeText(getApplicationContext(), "GPS Tidak aktif, tidak
    bisa melihat rute jalan indekos", Toast.LENGTH_SHORT).show();
    finish();
}

public void onStatusChanged(String provider, int status, Bundle extras){
}

```



```

private void ruteIndekos(Location loc){
    detailIndekos();
    if(loc!= null){
        lt1 = String.valueOf(loc.getLatitude()).toString();
        lg1 = String.valueOf(loc.getLongitude()).toString();

        webrute = (WebView)findViewById(R.id.webrute);
        webrute.getSettings().setJavaScriptEnabled(true);
        Log.d("Link",
"http://maps.google.com/maps?saddr="+lt1+", "+lg1+"&daddr="+lt+", "+lg);
        String uri =
"http://maps.google.com/maps?saddr="+lt1+", "+lg1+"&daddr="+lt+", "+lg;
        Intent intent = new
Intent(android.content.Intent.ACTION_VIEW, Uri.parse(uri));
        intent.setClassName("com.google.android.apps.maps",
"com.google.android.maps.MapActivity");
        startActivity(intent);

        //webrute.loadUrl("http://maps.google.com/maps?saddr="+lt+", "+lg+"&da
daddr="+lt1+", "+lg1);
    }
}

public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub
}
}

```