

TUGAS AKHIR

**PERANCANGAN DAN IMPLEMENTASI *TRY OUT* (UJI COBA)
SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) *ONLINE*
PADA ANDROID**

Disusun Oleh :

Nama : Kristian Ariyanto Zai

NIM : 2007-81-049



JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS ESA UNGGUL

JAKARTA

2012

TUGAS AKHIR

**PERANCANGAN DAN IMPLEMENTASI *TRY OUT* (UJI COBA)
SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) *ONLINE*
PADA ANDROID**

Diajukan untuk memenuhi Persyaratan Gelar Sarjana Komputer

Pada Jurusan Teknik Informatika Fakultas Ilmu Komputer

Universitas Esa Unggul

Disusun Oleh :

Nama : Kristian Ariyanto Zai

NIM : 2007-81-049



JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS ESA UNGGUL

JAKARTA

2012

LEMBAR PENGESAHAN TUGAS AKHIR

NAMA : Kristian Ariyanto Zai

NIM : 2007-81-049

JURUSAN : Teknik Informatika

FAKULTAS : Ilmu Komputer

JUDUL : PERANCANGAN DAN IMPLEMENTASI *TRY OUT*
(UJI COBA) SNMPTN (Seleksi Nasional Masuk
Perguruan Tinggi Negeri) *ONLINE* PADA ANDROID

Tugas Akhir ini telah disetujui sebagai syarat untuk memperoleh gelar Sarjana Komputer, Program Studi Strata I Ilmu Komputer Jurusan Teknik Informatika Universitas Esa Unggul.

Jakarta, Juli 2012

Mengetahui,

Ari Pambudi, SKom, M.Kom

Pembimbing Materi

Ir. I. Joko Dewanto, MM

Pembimbing Tulisan

Disetujui oleh,

Fransiskus Adikara, S.Kom, MMSi

Ketua Jurusan Teknik Informatika

Ari Pambudi, Skom, M.Kom

Dekan Fakultas Ilmu Komputer

LEMBAR PENGESAHAN SIDANG TUGAS AKHIR

Nama : Kristian Ariyanto Zai
Nim : 2007-81-049
Jurusan : Teknik Informatika
Program Studi : Ilmu Komputer
Judul Tugas Akhir : PERANCANGAN DAN IMPLEMENTASI *TRY OUT*
(UJI COBA) SNMPTN (Seleksi Nasional Masuk
Perguruan Tinggi Negeri) *ONLINE* PADA ANDROID

Dinyatakan LULUS mempertahankan Tugas Akhir pada ujian Tugas Akhir yang dilaksanakan di Universitas Esa Unggul.

Jakarta, 6 Februari 2012

Disetujui Oleh

Ketua Penguji : Ir. I. Joko Dewanto, MM : _____

Penguji I : Fransiskus Adikara, S.Kom., MMSI : _____

Penguji II : Riya Widayanti, Skom : _____

Mengetahui

Fransiskus Adikara, S.Kom., MMSI
Koordinator Tugas Akhir

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : **Kristian Ariyanto Zai**

Nim : **200781049**

Jurusan : **Teknik Informatika**

Menyatakan bahwa Tugas Akhir ini adalah murni hasil karya saya. Saya bersedia dikenakan sanksi pembuatan Tugas Akhir ini dan dicabut dari gelar Sarjana, apabila terbukti melakukan tindakan plagiat/mengutip dari karya orang lain.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, Desember 2012
Yang memberi pernyataan

Kristian Ariyanto Zai

ABSTRAK

Dalam persaingan masuk ke perguruan Tinggi Negri yang sangat ketat, dibutuhkan persiapan yang matang, salah satunya adalah dengan melakukan *Try Out* (Uji Coba), agar para peserta yang ingin mengikuti ujian SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) mendapatkan gambaran bagaimana ujian SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri), maka dengan itu dibutuhkan suatu aplikasi yang dapat membantu para peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) yang bersifat *mobile*.

Dengan melakukan persiapan melalui *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri), diharapkan para peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) dapat bersaing dan lolos SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri).

Diimplementasikan dalam sistem operasi *mobile* yaitu android yang sedang berkembang pesat, diharapkan para peserta yang mengikuti SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) dapat mempersiapkan diri dalam bersaing untuk lolos masuk Keperguruan Tinggi Negri.

Kata Kunci : *Try Out* (Uji Coba), SNMPTN, Android.

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa atas rahmat dan anugerahnya sehingga penulis dapat menyelesaikan penyusunan Skripsi ini. Judul Skripsi ini adalah PERANCANGAN DAN IMPLEMENTASI *TRY OUT* (UJI COBA) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) *ONLINE* PADA ANDROID. Penyusunan skripsi ini merupakan salah satu syarat untuk gelar kesarjanaan Jurusan Teknik Informatika Fakultas Ilmu Komputer Jenjang Pendidikan Stara – 1 pada Universitas Esa Unggul.

Penulis mengucapkan terima kasih kepada semua pihak yang telah terlibat dalam penyusunan skripsi ini, yaitu:

1. Bapak Ari Prambudi, S.Kom, M.Kom. selaku Dekan Fakultas Ilmu Komputer sekaligus sebagai pembimbing materi.
2. Bapak Fransiskus Adikara, S.Kom, MMSI. Selaku Ketua Jurusan Teknik Informatika.
3. Bapak Ir. I. Joko Dewanto, MM selaku pembimbing tulisan.
4. Seluruh Dosen dan Staff Fakultas Ilmu Komputer yang telah memberikan ilmu dan bimbingan kepada penulis selama masa perkuliahan.
5. Papah dan Mamah, yang telah memberikan dukungan baik secara materil, moral, dan doa.
6. Keempat kakak yang telah memberikan dukungan baik secara materil, moral, dan doa.
7. Teman-teman Gerakan Pemuda GPIB Obor Banten yang telah mendukung baik secara moral dan spritual kepada penulis.
8. Semua Mahasiswa Teknik Informatika angkatan 2007.

9. Buat Teman-teman GameBlok Team (Ashenk, Wawan, Manyun, Kentang, Deri) dan Fasilkom Bebas.
10. Buat Teman-teman 510 yang tidak bisa saya sebutkan satu persatu.
11. Dan yang terutama kepada Tuhan Yesus Kristus yang telah memberi saya kesehatan dan kekuatan jasmani maupun rohani selama penulisan skripsi.

Akhir kata, penulis berharap Skripsi ini dapat bermanfaat bagi para pembaca, terutama para pembaca yang ingin melakukan penelitian di bidang yang sama.

Tangerang, Juli 2012

Penulis
(Kristian Ariyanto Zai)

DAFTAR ISI

HALAMAN JUDUL LUAR.....	i
HALAMAN JUDUL DALAM.....	ii
HALAMAN PERSETUJUAN TUGAS AKHIR.....	iii
ABSTRAK.....	iv
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	6
2.1 Pengertian Perguruan Tinggi Negri.....	6
2.2 Pengertian SNMPTN.....	7
2.3 Pengertian Android.....	7

2.3.1	Versi Android..	8
2.3.2	Kelebihan Android.....	10
2.3.3	Arsitektur Android.....	12
2.3.4	Fundamental Aplikasi.....	15
	2.3.4.1 <i>Activitie</i>	15
	2.3.4.2 <i>Service</i>	16
	2.3.4.3 <i>Broadcast Receiver</i>	16
	2.3.4.4 <i>Content Provider</i>	17
2.3.5	<i>The Dalvik Virtual Machine</i>	17
2.4	PHP Cake.....	18
2.5	MySQL.....	19
2.6	JAVA.....	20
2.7	<i>Android SDK Manager dan AVD Manager</i>	20
2.8	Eclipse.....	21
2.9	<i>Android Development Tools (ADT)</i>	21
2.10	<i>UML (Unified Modeling Language)</i>	22
	2.10.1 <i>Use Case Diagram</i>	22
	2.10.2 <i>Class Diagram</i>	24
	2.10.3 <i>Statechart Diagram</i>	26
	2.10.4 <i>Sequence Diagram</i>	27
	2.10.5 <i>Activity Diagram</i>	28
2.11	Kerangka Pemikiran.....	30
BAB III GAMBARAN UMUM PENELITIAN.....		32
3.1	Proses Bisnis.....	32
	3.1.1 Jalur Masuk Perguruan Tinggi Negeri.....	32
	3.1.2 Jenis Ujian SNMPTN.....	34
	3.1.3 Sistem Penilaian SNMPTN.....	36

3.1.4	<i>Try Out</i> (Uji Coba) SNMPTN.....	38
3.2	Analisis Masalah.....	38
3.3	Rencana Solusi Pemecahan.....	40
BAB IV	ANALISIS DAN PERANCANGAN SISTEM.....	41
4.1	Metode Perancangan Sistem.....	41
4.2	Perancangan Sistem.....	42
4.2.1	<i>Use Case Diagram</i>	43
4.2.2	<i>Class Diagram</i>	45
4.2.3	<i>Sequence diagram</i>	46
4.2.3.1	<i>Sequence Diagram Try Out</i> (Uji Coba).....	47
4.2.3.2	<i>Sequence Diagram View Nilai</i>	47
4.2.3.3	<i>Sequence Diagram Manage Ujian</i>	47
4.2.3.4	<i>Sequence Diagram Manage Soal</i>	48
4.2.3.5	<i>Sequence Diagram Manage Jawaban</i>	49
4.2.3.6	<i>Sequence Diagram Manage List Jawaban</i>	50
4.2.3.7	<i>Sequence Diagram Manage Nilai</i>	51
4.2.3.8	<i>Sequence Diagram Manage User</i>	52
4.2.4	<i>Collaboration Diagram</i>	53
4.2.4.1	<i>Calaboration Diagram Try Out</i> (Uji Coba)...	53
4.2.4.2	<i>Calaboration Diagram View Nilai</i>	54
4.2.4.3	<i>Calaboration Diagram Manage Ujian</i>	55
4.2.4.4	<i>Calaboration Diagram Manage Soal</i>	55
4.2.4.5	<i>Calaboration Diagram Manage Jawaban</i>	56
4.2.4.6	<i>Calaboration Diagram Manage List Jawaban</i>	57
4.2.4.7	<i>Calaboration Diagram Manage Nilai</i>	58
4.2.4.8	<i>Calaboration Diagram Manage User</i>	58
4.2.5	<i>Activity Diagram</i>	59
4.2.6	<i>Statechart Diagram</i>	62

4.2.6.1 <i>Statechart Diagram Try Out (Uji Coba)</i>	63
4.2.6.2 <i>Statechart Diagram View Nilai</i>	63
4.2.6.3 <i>Statechart Diagram Manage Ujian</i>	64
4.2.6.4 <i>Statechart Diagram Manage Soal</i>	65
4.2.6.5 <i>Statechart Diagram Manage Jawaban</i>	65
4.2.6.6 <i>Statechart Diagram Manage List Jawaban</i>	66
4.2.6.7 <i>Statechart Diagram Manage Nilai</i>	67
4.2.6.8 <i>Statechart Diagram Manage User</i>	68
4.2.7 <i>Development Diagram</i>	68
BAB V SIMPULAN DAN SARAN.....	69
5.1 Simpulan.....	69
5.2 Saran.....	69
DAFTAR PUSTAKA.....	71
IDENTITAS PENYUSUN.....	72
LAMPIRAN	73

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Android.....	12
Gambar 2.2 Arsitektur CakePHP.....	19
Gambar 2.3 Kerangka Pemikiran.....	30
Gambar 3.1 Pendaftaran SNMPTN	34
Gambar 4.1 <i>Use Case</i> Diagram.....	44
Gambar 4.2 Class Diagram.....	45
Gambar 4.3 <i>Sequence</i> Diagram <i>Try Out</i> (Uji Coba).....	46
Gambar 4.4 <i>Sequence</i> Diagram <i>View</i> Nilai.....	47
Gambar 4.5 <i>Sequence</i> Diagram <i>Manage</i> Ujian.....	48
Gambar 4.6 <i>Sequence</i> Diagram <i>Manage</i> Soal.....	49
Gambar 4.7 <i>Sequence</i> Diagram <i>Manage</i> Jawaban.....	50
Gambar 4.8 <i>Sequence</i> Diagram <i>Manage</i> List Jawaban	51
Gambar 4.9 <i>Sequence</i> Diagram <i>Manage</i> Nilai.....	52
Gambar 4.10 <i>Sequence</i> Diagram <i>Manage</i> <i>User</i>	53
Gambar 4.11 <i>Calloboratin</i> Diagram <i>Try Out</i>	54
Gambar 4.12 <i>Calloboratin</i> Diagram Nilai.....	54
Gambar 4.13 <i>Calloboratin</i> Diagram <i>Manage</i> Ujian.....	55
Gambar 4.14 <i>Calloboratin</i> Diagram <i>Manage</i> Soal.....	56
Gambar 4.15 <i>Calloboratin</i> Diagram <i>Manage</i> Jawaban.....	56
Gambar 4.16 <i>Calloboratin</i> Diagram <i>Manage</i> List Jawaban.....	57
Gambar 4.17 <i>Calloboratin</i> Diagram <i>Manage</i> Nilai.....	58
Gambar 4.18 <i>Calloboratin</i> Diagram <i>Manage</i> <i>User</i>	58
Gambar 4.19 <i>Activity</i> Diagram <i>Manage</i> Ujian.....	59
Gambar 4.20 <i>Activity</i> Diagram <i>Manage</i> Soal.....	60
Gambar 4.21 <i>Activity</i> Diagram <i>Manage</i> Jawaban.....	60

Gambar 4.22 <i>Activity Diagram Manage List Jawaban</i>	61
Gambar 4.23 <i>Activity Diagram Manage Nilai</i>	61
Gambar 4.24 <i>Activity Diagram Manage User</i>	62
Gambar 4.24 <i>Activity Diagram Try Out (Uji Coba)</i>	63
Gambar 4.25 <i>Statechart Diagram View Nilai</i>	64
Gambar 4.26 <i>Statechart Diagram Manage Ujian</i>	64
Gambar 4.27 <i>Statechart Diagram Manage Ujian</i>	65
Gambar 4.28 <i>Statechart Diagram Manage Jawaban</i>	66
Gambar 4.29 <i>Statechart Diagram Manage List Jawaban</i>	66
Gambar 4.30 <i>Statechart Diagram Manage Nilai</i>	67
Gambar 4.31 <i>Statechart Diagram Manage User</i>	68
Gambar 4.33 <i>Deployment Diagram</i>	68

DAFTAR TABEL

Table 2.1 <i>Use Case Diagram</i>	23
Tabel 2.2 <i>Clas Diagram</i>	25
Tabel 2.3 <i>Statechart Diagram</i>	26
Tabel 2.4 <i>Sequence diagram</i>	27
Tabel 2.5 : <i>Activity Diagram</i>	28
Tabel 3.1 Test Bidang Studi Dasar.....	35
Tabel 3.2 Tabel Test Bidang Studi (TBS) IPA.....	35
Tabel 3.3 Test Bidang Studi (TBS) IPS.....	35
Tabel 3.4 Tabel Simulasi Penilaian Tes Bidang Studi Prediktif.....	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Memasuki era globalisasi, pendidikan formal sangat penting bagi setiap orang. Pendidikan yang bersifat sistematis, berstruktur dan bertingkat dimulai dari sekolah dasar sampai keperguruan tinggi. Pendidikan formal merupakan bagian dari pendidikan nasional yang bertujuan untuk membentuk seseorang yang dapat bersaing di era global saat ini.

Dalam pendidikan formal yang terstruktur dan berjenjang terdiri dari pendidikan dasar, pendidikan menengah, dan pendidikan tinggi. Pemilihan tempat menjalani pendidikan sangatlah penting, terlebih dalam kenaikan jenjang pendidikan dari jenjang SMA/ sederajat keperguruan tinggi. Sebagian siswa yang baru lulus SMA/ sederajat cenderung memilih perguruan tinggi negeri dibandingkan dengan perguruan tinggi swasta. Hal ini dikarenakan biaya yang lebih terjangkau, selain itu dari segi kualitas pendidikan perguruan tinggi negeri juga lebih baik dengan adanya tenaga pengajar yang handal dan berkompeten dibidangnya, dengan begitu diharapkan lulusan perguruan tinggi negeri mempunyai kualitas dan dapat bersaing di era global saat ini.

Namun kapasitas perguruan tinggi negeri sangat terbatas, membuat tidak semua siswa yang mendaftar dapat masuk seluruhnya. Oleh karena itu para calon mahasiswa perguruan tinggi negeri harus mengikuti seleksi penerimaan mahasiswa yang dinamakan SNMPTN yakni Seleksi Nasional Masuk Perguruan Tinggi Negeri. Dalam SNMPTN para siswa yang ingin masuk keperguruan tinggi negeri harus bersaing untuk dapat masuk keperguruan tinggi negeri.

Dengan persaingan yang ketat, para siswa yang ingin masuk keperguruan tinggi negeri harus mempersiapkan diri agar dapat bersaing dan masuk keperguruan tinggi negeri yang rencanakan, mulai dari pemilihan jurusan, pemahaman penilaian dan peraturan, adaptasi terhadap soal-soal yang diujikan dan mengikuti *Try Out*. Persiapan tersebut dapat dengan mudah kita dapatkan ditempat bimbingan belajar atau yang biasa disebut dengan bimbel. Namun mengikuti bimbingan belajar atau bimbel membutuhkan biaya, selain itu para peserta bimbel juga harus mengikuti jadwal yang sudah ditentukan oleh penyelenggara bimbel, sehingga bimbingan belajar atau bimbel masih kurang efektif dari segi biaya, waktu dan tempat.

Seiring dengan perkembangan zaman baik dari segi teknologi maupun dari segi lainnya, khususnya dibidang pendidikan maka dibutuhkan suatu terobosan–terobosan atau inovasi–inovasi yang dapat memudahkan pelajar dalam melakukan *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri). Maka muncul sebuah gagasan untuk membuat *Try Out* (Uji coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) yang bersifat *mobile*. Salah satu perangkat *mobile* yang kemajuannya sangat pesat adalah *handphone*, dimana yang dulu kegunaannya hanya untuk komunikasi, sekarang *handphone* dapat berguna layaknya seperti komputer yang dapat di install berbagai macam aplikasi.

Maka dengan munculnya masalah tersebut, disini penulis mencoba membuat sesuatu yang bisa digunakan/diterapkan dalam kehidupan sehari–hari yaitu **PERANCANGAN DAN IMPLEMENTASI TRY OUT (UJI COBA) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) ONLINE PADA ANDROID.**

1.2 Identifikasi Masalah

Berdasarkan uraian di atas maka dapat terlihat suatu permasalahan. Bagaimana membangun dan merealisasikan suatu aplikasi yang bersifat *mobile* yang berguna untuk membantu siswa dalam melakukan *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) yang dapat dilakukan di mana saja dan kapan saja, sehingga para siswa yang ingin mengikuti SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dapat lebih siap dalam mengikuti SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

1.3 Tujuan Penelitian

- 1) Mengidentifikasi bagaimana sistem SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).
- 2) Membangun sebuah aplikasi perangkat lunak *mobile* yang diharapkan dapat membantu dalam *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).
- 3) Membantu seseorang dalam melakukan *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

1.4 Manfaat Penelitian

- 1) Mempermudah seseorang yang ingin melakukan *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) khususnya bagi para pengguna android.
- 2) Agar proses *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dapat dilakukan dimana saja, kapan saja.

1.5 Batasan Masalah

Agar penelitian ini tidak melebar pembahasannya, maka diperlukan pembatasan masalah. Batasan masalahnya adalah :

- 1) Penelitian ini hanya mengolah data tentang *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).
- 2) Sistem yang dibangun dapat mengupdate data-data atau soal-soal disimpan dalam database.
- 3) Sistem *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) akan berjalan pada sistem oprasi android.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini diuraikan dalam lima bab dan isinya akan diuraikan sebagai berikut :

BAB I PENDAHULUAN

Dalam bab ini akan dibahas mengenai latar belakang penelitian, rumusan masalah, ruang lingkup, tujuan dan manfaat penelitian.

BAB II LANDASAN TEORI

Dalam bab ini akan diuraikan mengenai teori-teori yang mendukung yang berhubungan dengan penelitian ini.

BAB III ANALISIS SISTEM YANG BERJALAN

Dalam bab ini dibahas mengenai latar belakang, sejarah, struktur organisasi serta gambaran sistem yang sedang berjalan.

BAB IV SISTEM YANG DIUSULKAN

Dalam bab ini akan dijelaskan mengenai sistem yang diusulkan oleh penulis untuk dapat dipertimbangkan.

BAB V KESIMPULAN DAN SARAN

Dalam bab ini akan dijelaskan kesimpulan dari skripsi ini dan saran yang diberikan oleh penulis berdasarkan sistem yang diusulkan agar dapat dipertimbangkan oleh perusahaan.

BAB II

LANDASAN TEORI

2.1 Pengertian Perguruan Tinggi

Perguruan tinggi adalah satuan pendidikan penyelenggara pendidikan tinggi. Peserta didik perguruan tinggi disebut mahasiswa, sedangkan tenaga pendidik perguruan tinggi disebut dosen. Di Indonesia, perguruan tinggi dapat berbentuk akademik, institut, politeknik, sekolah tinggi, dan universitas. Perguruan tinggi dapat menyelenggarakan pendidikan akademik, profesi, dan vokasi dengan program pendidikan diploma (D1,D2,D3,D4), sarjan (S1), Megister (S2), doktor (S3) dan spesialis.

Sebuah universitas negeri adalah sebuah universitas yang didanai oleh pemerintah nasional atau daerah, berlawanan dari universitas swasta. Sebuah universitas nasional dapat atau tidak dapat dianggap sebagai universitas negeri, tergantung daerahnya.

Perguruan tinggi berfungsi memberikan pelayanan pendidikan tinggi kepada mahasiswa, melakukan pengembangan ilmu, teknologi, dan seni melalui penelitian ilmiah, yang hasilnya didiseminasikan melalui proses pembelajaran pada pendidikan tinggi serta diabdikan untuk pengembangan masyarakat.

Menurut jenisnya, perguruan tinggi dibagi menjadi dua yaitu

- 1) Perguruan tinggi negeri adalah perguruan tinggi yang diselenggarakan oleh pemerintah.
- 2) Perguruan tinggi swasta adalah perguruan tinggi yang diselenggarakan oleh pihak swasta.

2.2 Pengertian SNMPTN

SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) adalah sebuah pola seleksi penerimaan yang dilaksanakan secara bersama oleh seluruh perguruan tinggi negeri dalam suatu sistem yang terpadu dan dilaksanakan secara serentak yang disepakati oleh Pengurus Majelis Rektor Perguruan Tinggi Negeri Indonesia. SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dilaksanakan melalui dua jalur yaitu jalur undangan, dan jalur ujian tertulis. Hal tersebut bertujuan untuk memangkas peminat yang ingin masuk keperguruan tinggi negeri yang memiliki keterbatasan kapasitas dalam menampung mahasiswa baru.

2.3 Pengertian Android

Menurut (Nazrudin, 2011) Android adalah sistem operasi untuk *mobile device* yang awalnya dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Android dibuat berdasarkan kernel Linux yang dimodifikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Android dibangun menggunakan bahasa Java. Untuk tampilan dapat juga dibangun menggunakan file XML Android. Android menyediakan android SDK yang dapat dengan mudah dipadukan dengan Eclipse sebagai *Integrated*

Development Environment. *SQLite* dapat berdiri sendiri tanpa membutuhkan kehadiran server.

2.3.1 Versi Android

1) versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2) versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3) versi 2.0/2.1 (*Eclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru.

4) versi 2.2 (*Froyo: Frozen Yoghurt*)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

5) versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan, peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization, dan bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

6) versi 3.0/3.1 (*Honeycomb*)

Android *Honeycomb* dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

7) Versi 4.0 (*Ice Cream Sandwich*)

Android *Ice Cream Sandwich* yang dirilis pada 19 oktober 2011 membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan data dan kontrol

terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan NFC

2.3.2 Kelebihan Android

Menurut (Nazrudin, 2011) Sudah banyak *platform* untuk perangkat selular, termasuk didalamnya Symbian, iPhone, Windows *Mobile*, BlackBerry, Java *Mobile Edition*, Linux *Mobile* (LiM0), dan banyak lagi. Namun ada beberapa yang menjadi kelebihan Android. Walaupun beberapa fitur yang ada telah muncul sebelumnya pada *platform* lain, Android adalah yang pertama menggabungkan hal seperti berikut :

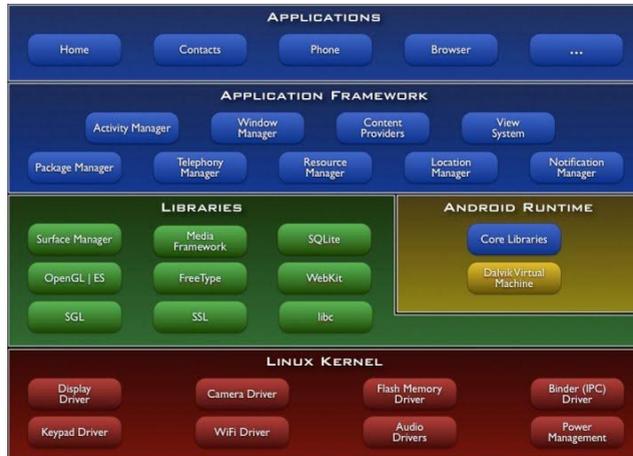
- 1) Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasis Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai keinginan tanpa harus memikirkan biaya royalti. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh segala *vendor*.
- 2) Arsitektur komponen dasar Android terinspirasi dari teknologi internet Mashup. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti

dengan menggunakan komponen lain yang sesuai dengan aplikasi yang akan dikembangkan.

- 3) Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database SQL*, *browser* dan penggunaan peta. Semuanya sudah tertanam pada Android sehingga memudahkan dalam mengembangkan aplikasi.
- 4) Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja system menjadi lebih stabil. Pengguna tak perlu cemas dalam menggunakan aplikasi pada perangkat yang memori yang terbatas.
- 5) Dukungan grafis terbaik, dengan adanya dukungan *2D* grafis dan animasi yang diilhami oleh *Flash* menyatu dalam *3D* menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
- 6) Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program dibangun menggunakan bahasa pemrograman Java dan dieksekusi oleh mesin *virtual* Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan Keyboard, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.3.3 Arsitektur Android

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut.



Gambar 2.1 Arsitektur Android

Sumber : <http://developer.android.com/guide/basics/what-is-android.html>

1) *Application* dan *widjets*

Menurut (Nazrudin, 2011) *Application* dan *widjets* ini adalah layer dimana kita berhubungan dengan palikasi saja, di mana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program sms, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan pemrograman Java

2) Applications Frameworks

Menurut (Nazrudin, 2011) Android adalah “*Open Development Platform*” yaitu android menawarkan bagi pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembangan bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarem, dan menambah status *notification*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan.

Sehingga bias kita simpulkan *applications frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem oprasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon.

Yang termasuk komponen-komponen *Applications Frameworks* sebagai berikut

- (1) *Views*
- (2) *Content Provider*
- (3) *Resource Manager*
- (4) *Notification Manager*
- (5) *Activity Manager*

3) *Libraries*

Menurut (Nazrudin, 2011) *Libraries* ini adalah layer dimana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library* inti seperti Libe dan SSL, serta :

- (1) Media untuk pememutaran audio dan video.
- (2) Manajemen tampilan
- (3) *Graphics* mencakup *SGL* dan *OpenGL* untuk grafis *2D* dan *3D*.
- (4) *SQLite* untuk dukungan database.
- (5) *SSL* dan *Webkit* terintegrasi dengan *web browser* dan *security*.
- (6) *liveWebcore* dengan mencakup modern *web browser* dengan *engine embedded web view*.

4) Android Run Time

Menurut (Nazrudin, 2011) Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi android. di dalam android *Run Time* dibagi menjadi dua bagian yaitu :

- (1) *Core Libraris* : Aplikasi android dibangun dengan bahasa Java, sementara Dalvik sebagai virtual mesinnya, bukan *Virtual Machine Java*, sehingga diperlukan sebuah

libraries yang berfungsi untuk menterjemahkan bahasa Java yang ditangani oleh *Core Libraris*.

(2) *Dalvic Virtual Machine* : Virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5) Linux Kernel

Menurut (Nazrudin, 2011) Linux Kernel adalah *layer* dimana inti dari oprating sistem dari android itu berada. Berisi file-file sistem yang mengatur *system processing, memory, resource, drivers*, dan sistem-sistem oprasi android lainnya.

2.3.4 Fundamental Aplikasi

Menurut (Nazrudin, 2011) Aplikasi android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data file *resource* yang dibutuhkan oleh aplikasi, di mana prosesnya di-package oleh tools yang dinamakan “*apt tool*” ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang kita sebut aplikasi, dan intinya dapat di instal di perangkat *mobile*.

2.3.4.1 Activities

Menurut (Nazrudin, 2011) Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk

menampilkan aplikasi atau yang bertindak sebagai *user interface (UI)* saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain kita dapat melakukannya dengan satu *event*, misalnya *click* tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan method *Activity setContentView()*. *ContentView* adalah objek yang berada pada root hirarki.

2.3.4.2 Service

Menurut (Nazrudin, 2011) *Service* tidak memiliki *Graphic User Interface (GUI)*, tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya, media player sedang memutar lagu dari *list* yang ada, aplikasi ini memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu, atau menulis sms sambil *player* sedang jalan. Untuk menjaga musik tetap dijalankan pada thread utama dari proses aplikasi.

2.3.4.3 Broadcast Receiver

Menurut (Nazrudin, 2011) *Broadcast Receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu berubah, baterai lemah, gambar telah selesai diambil camera atau pengubahan referensi bahasa yang digunakan. *Broadcast Receiver* tidak memiliki *User*

Interface (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang meraka terima, atau *notification manager* untuk member tahu kepada pengguna, seperti lampu latar atau *vibrating* perangkat.

2.3.4.4 Content Provider

Menurut (Nazrudin, 2011) *Content Provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpulkan dalam file sistem seperti *database SQLite*. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (*Map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi *Content Provider*.

2.3.5 The Dalvik Virtual Machine

Menurut (Nazrudin, 2011) Salah satu elemen kunci android adalah *Dalvik Virtual Machine (DVM)*. Android berjalan di dalam *Dalvik Virtual Machine (DVM)* bukan di *Java Virtual Machine (JVM)*, meskipun berbeda namun banyak juga persamaan dengan *Java Virtual Machine (JVM)* seperti *Java ME (Java Mobile Edition)*, tetapi android menggunakan *Virtual Machine* sendiri.

Dalvik Virtual Machine (DVM) adalah “*register bases*” sementara *Java Virtual Machine (JVM)* adalah “*stack based*”, *DVM* didesain dan ditulis oleh Dan Bornsten dan beberapa *engineers* Google lainnya. *Dalvik Virtual Machine (DVM)*

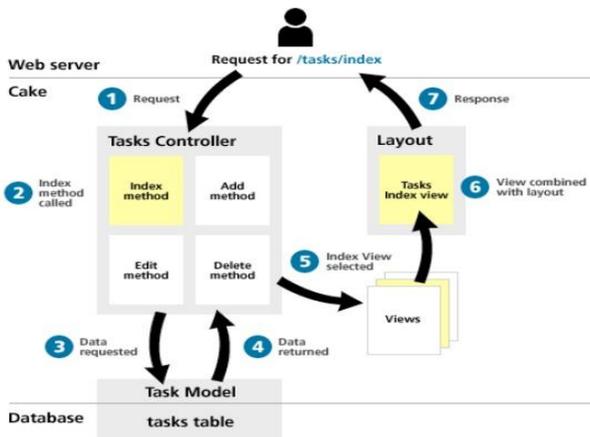
menggunakan Kernel Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori.

Semua *hardware* yang berbasis android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil.

2.4 *PHP Cake*

PHP Cake adalah adalah sebuah framework atau kerangka kerja untuk membuat aplikasi CRUD (*Create, Read, Update, Delete*) berbasis bahasa pemrograman PHP. *CakePHP* juga menjadi salah satu framework pilihan yang memungkinkan seorang developer web untuk membuat sebuah aplikasi dengan karakter pengembangan RAD (*Rapid Application Development*), yang memungkinkan untuk digunakan dan dikembangkan menjadi aplikasi lain yang lebih kompleks. *CakePHP* masih satu trah dan mempunyai hubungan darah yang cukup erat dengan Ruby on Rails, sebuah *framework* pemrograman Ruby. Berikut adalah kelebihan yang dimiliki *PHPcake*.

Struktur kerja CakePHP



Gambar 2.2 CakePHP
Sumber : <http://te.ugm.ac.id>

Dari gambar di atas, cara kerja CakePHP sebagian besar sudah bisa dijelaskan. Pertama kali, user melakukan request, yang akan ditangani oleh controller. Controller memiliki banyak metode yang akan menangani request. Controller akan mengambil data pada database melalui model. Model memberikan data yang relevan kepada controller. Controller akan mengolah data dan memberikannya pada view (tampilan) yang dipilih. Tampilan inilah yang akan dilihat oleh user pada browser-nya.

2.5 MySQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (Structure Query Language). MySQL pertama kali dirilis oleh seorang programmer database bernama Michael Widenius. Selain database server, MySQL juga merupakan program yang

dapat mengakses suatu database MySQL yang berposisi sebagai server, yang berarti program kita sebagai client sehingga MySQL adalah sebuah database yang dapat digunakan sebagai client maupun server.

2.6 JAVA

Menurut (John Byus, 1988) Java adalah salah satu bahasa pemrograman yang dapat dijalankan diberbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintak yang terdapat pada C dan C++ namun dengan sintak model objek yang lebih sederhana. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa *platform* sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web*.

2.7 Android SDK Manager dan AVD Manager

Menurut (Nazrudin, 2011) Android SDK Manager merupakan sebuah alat pengembangan perangkat lunak yang berguna untuk mengembangkan dan membuat aplikasi untuk *platform* Android. Didalam Android SDK terdapat sebuah project sample dengan kode sumber, alat-alat untuk membuat aplikasi, sebuah *emulator* Android (AVD), dan perpustakaan yang dibutuhkan untuk membangun aplikasi Android. Aplikasi yang ditulis pada bahasa pemrograman Java dan berjalan di

Dalvik, mesin virtual yang dirancang khusus untuk penggunaan embedded yang berjalan di kernel Linux.

2.8 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak juga dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat-sifat dari Eclipse:

- 1) Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- 2) Multi-language: Eclipse dikembangkan menggunakan bahasa pemrograman Java, tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- 3) *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse juga bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain-lain.

2.9 Android Development Tools (ADT)

Menurut (Nazrudin, 2011) *Android Development Tools (ADT)* merupakan plug in Eclipse IDE yang dirancang untuk memberi kemudahan dalam membuat aplikasi Android.

ADT memperluas kemampuan Eclipse untuk memberi kemudahan kepada seseorang agar membuat project Android, membuat aplikasi UI, menambahkan komponen berdasarkan *Android Framework API*, dan *debug* aplikasi menggunakan *Android SDK*.

Mengembangkan Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai pembuatan aplikasi pada

sistem operasi Android. Dengan setup project yang telah disediakan, dan integrasi peralatan, custom XML editor, dan debug panel output. ADT memberikan dorongan luar biasa untuk mengembangkan aplikasi Android. Dengan setup project yang telah disediakan, serta integrasi peralatan, custom XML editor, dan debug panel output.

2.10 UML (Unified Modeling Language)

Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang mengizinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagai dan mengkonsumsi rancangan mereka dengan yang lain.

2.10.1 Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem untuk mencapai tujuan tertentu. Setiap scenario menjelaskan urutan kejadian. *Use case* adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna. *Use case* biasanya menggunakan *actors*. *Actors* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Table 2.1 Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Sesuatu yang menghubungkan antara objek satu dengan objek lainnya.

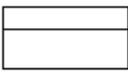
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : www.pribadiraharja.com

2.10.2 *Class Diagram*

Class diagram adalah bagian dari UML yang menggambarkan sebuah kumpulan dari kelas-kelas yang ada dan hubungan diantara kelas tersebut dimana setiap kelas mempunyai attributes dan operations.

Tabel 2.2 Clas Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.10.3 Statechart Diagram

Statechart Diagram adalah bagian dari UML yang menggambarkan tingkah laku yang umum dari sebuah objek didalam sebuah class yang spesifik dan berisi states dan transisi diantaranya.

Tabel 2.3 Statechart Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	menghubungkan antara objek satu dengan objek lainnya.

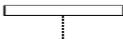
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
---	---	-------------	--

Sumber : www.pribadiraharja.com

2.10.4 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh objek dan message (pesan) yang diletakkan diantara objek-objek ini di dalam use case.

Tabel 2.4 Sequence diagram

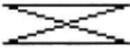
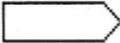
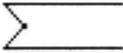
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.10.5 Activity Diagram

Activity Diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai peran seperti halnya a flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku parallel sedangkan flowchart tidak bisa. Selain sebagai gambaran detail sebuah use case diagram, activity diagram bisa juga untuk menjabarkan suatu state tertentu dan statechart diagram dimana fungsinya untuk menerangkan dan mendeskripsikan internal behavior suatu metode/state dan menunjukkan aliran action yang di kendalikan (driven by) oleh action sebelumnya.

Tabel 2.5 : Activity Diagram

NO	Simbol	Keterangan
1		Titik awal
2		Titik akhir
3		<i>Activity</i>

4		Pilihan untuk mengambil keputusan
5		<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
6		Rake menunjukkan bahwa adanya dekomposisi
7		Tanda waktu
8		tanda pengiriman dan urutan aktifitas dalam satu proses
9		Tanda Penerimaan
10		Aliran akhir (flow final)

Sumber : www.pribadiraharja.com

2.11 Kerangka Pemikiran



Gambar 2.3 Kerangka Pemikiran

1. Dari gambar di atas, yang pertama dilakukan adalah mengidentifikasi sistem SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) yang berkaitan dengan aplikasi yang akan dibuat.
2. Setelah mengidentifikasi, yang akan dilakukan adalah pemodelan visual yaitu dengan menggunakan UML (*Unified Modeling Language*). Hal tersebut bertujuan untuk mempermudah pada saat pembuatan sistem pada sebuah aplikasi pelatihan SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

3. Langkah berikutnya adalah pengembangan system. Sistem yang akan dibangun adalah sistem yang akan berjalan di sistem operasi android, dan menggunakan admin yang bertugas memanege sistem *Try Out* (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri)
4. Setelah itu menguji aplikasi yang sudah dibangun.
5. Lalu yang terakhir kesimpulan dan saran.

BAB III

GAMBARAN UMUM OBJEK PENELITIAN

3.1 Proses Bisnis

Setelah lulus SMA/Sederajat para siswa banyak yang ingin melanjutkan keperguruan tinggi. Banyak para siswa yang memilih perguruan tinggi negeri menjadi pilihan utama hal tersebut dikarenakan biaya yang lebih terjangkau, selain itu dari segi kualitas pendidikan perguruan tinggi negeri juga lebih baik dengan adanya tenaga pengajar yang handal dan berkompeten dibidangnya, dengan begitu diharapkan lulusan perguruan tinggi negeri mempunyai kualitas dan dapat bersaing di era global saat ini.

SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) adalah sebuah pola seleksi yang dilaksanakan secara bersama oleh seluruh perguruan tinggi negeri dalam suatu sistem yang terpadu dan dilaksanakan secara serentak. Hal tersebut bertujuan untuk memangkas peminat yang ingin masuk keperguruan tinggi negeri yang memiliki keterbatasan kapasitas dalam menampung mahasiswa baru.

3.1.1 Jalur Masuk Perguruan Tinggi Negeri

SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dilaksanakan melalui dua jalur yaitu jalur undangan dan jalur ujian tertulis.

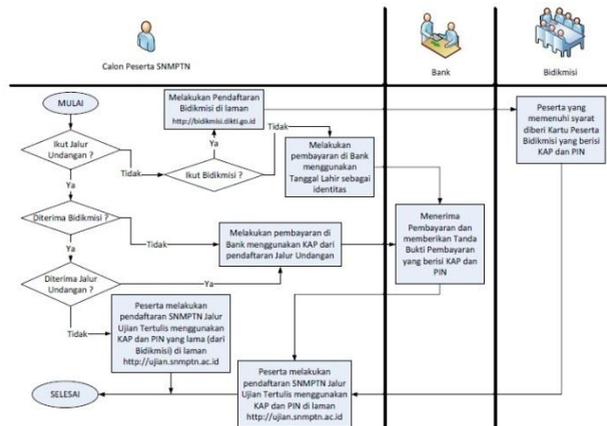
1) Jalur Undangan

Penerimaan mahasiswa baru harus memenuhi prinsip adil, transparan, dan tidak diskriminatif dengan tidak membedakan jenis kelamin, agama, suku, ras, kedudukan sosial, dan tingkat kemampuan ekonomi calon mahasiswa dengan tetap memperhatikan potensi calon mahasiswa dan kekhususan perguruan tinggi. Perguruan tinggi sebagai penyelenggara pendidikan setelah SMA/SMK/MA/MAK hendaknya dapat menerima calon mahasiswa yang berprestasi akademik tinggi dan diprediksi akan berhasil menyelesaikan studi di perguruan tinggi berdasarkan penilaian dan rekomendasi Kepala Sekolah. Siswa yang berprestasi tinggi dan secara konsisten menunjukkan prestasinya tersebut layak mendapatkan kesempatan untuk menjadi calon mahasiswa. Sekolah sebagai satuan pendidikan dan guru sebagai pendidik diyakini selalu menjunjung tinggi kehormatan dan kejujuran sebagai bagian dari prinsip pendidikan berkarakter. Dengan demikian sekolah dapat diberikan penghargaan dan kepercayaan melakukan seleksi awal calon mahasiswa yang berprestasi akademik dan diharapkan dapat menyelesaikan pendidikan tinggi dengan baik.

2) Jalur Ujian Tertulis

Jalur Ujian Tertulis dan Keterampilan merupakan salah satu jalur SNMPTN yang diharapkan mampu memprediksi keberhasilan calon mahasiswa menyelesaikan studi tepat waktu di perguruan tinggi. Ujian tertulis menggunakan soal ujian yang dikembangkan sedemikian rupa sehingga memenuhi persyaratan validitas, tingkat kesulitan, dan daya pembeda yang memadai. Disamping itu, soal SNMPTN dirancang untuk mengukur kemampuan umum yang diprediksi menentukan keberhasilan

calon mahasiswa di semua program studi, yakni kemampuan berpikir tingkat tinggi (*higher order thinking*), yang meliputi Tes Potensi Akademik (TPA), Tes Bidang Studi Dasar, Tes Bidang Studi IPA, dan Tes Bidang Studi IPS. Ujian Keterampilan dilakukan untuk mengukur kemampuan dalam bidang ilmu Seni dan Keolahragaan.



Gambar 3.1 Pendaftaran SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri)

3.1.2 Jenis Ujian SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri)

Dalam ujian SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) terdapat dua ujian yaitu ujian tertulis dan ujian keterampilan.

1) Ujian Tertulis

Ujian tertulis terdiri dari

- (1) Test Potensi Akademik (TPA)
- (2) Test Bidang Studi (TBS)

Tabel 3.1 Test Bidang Studi Dasar

No	Test Bidang Studi (TBS)
1	Matematika Dasar
2	Bahasa Indonesia
3	Bahasa Inggris

Tabel 3.2 Tabel Test Bidang Studi (TBS) IPA

No	Test Bidang Studi (TBS) IPA
1	Matematika
2	Biologi
3	Kimia
4	Fisika

3.3 Tabel Test Bidang Studi (TBS) IPS

No	Test Bidang Studi (TBS) IPS
1	Sosiologi
2	Sejarah
3	Geografi
4	Ekonomi

2) Ujian Keterampilan

Ujian Keterampilan diperuntukan bagi peminat program studi bidang ilmu seni dan olahraga.

3.1.3 Sistem Penilaian SMNPTN (Seleksi Masuk Perguruan Tinggi Negeri)

Sistem penilaian SNMPTN Yang diberlakukan adalah sistem penilaian PERSENTIL yaitu penilaian per mata pelajaran yang diujikan secara terpisah. Dengan sistem persentil peserta SNMPTN harus mengerjakan semua mata pelajaran yang diujikan karena semua hasil tes akan diperiksa dan dinilai secara terpisah. Penggunaan sistem penilaian tersebut mempunyai dampak yang berbeda dalam menentukan siapa yang akan lolos SNMPTN.

Sebelum sistem penilaian PERSENTIL, SNMPTN menggunakan sistem penilaian NILAI MENTAH banyak peserta yang lolos atau diterima oleh satu program studi/jurusan di PTN dengan hanya mengandalkan satu mata pelajaran yang dianggap paling dikuasainya. Sedangkan mata pelajaran lain tidak dijawab satupun dalam lembar jawaban (kosong). Oleh sebab itu banyak peserta/siswa dari Jurusan IPA lolos (diterima) di Akuntansi UI atau Fikom Unpad, karena mereka mengandalkan kemampuan di matematika, sementara pelajaran Ekonominya jelek bahkan mungkin tidak diisi/dikosongkan untuk menghindari nilai minus. Namun demikian bukan berarti yang lolos SNMPTN adalah peserta yang mengisi seluruh mata pelajaran dengan 100% benar jawabannya.

Perlu dipahami pula bahwa setiap bidang studi (mata pelajaran) Tes Bidang Studi Prediktif (TBSP) di nilai berdasarkan aturan, sebagai berikut :

- 1) Apabila jawaban betul dikalikan 4 (empat)
- 2) Apabila jawaban salah dikali -1. (minus satu),
- 3) Apabila tidak dijawab dikalikan 0 (nol).

Dari aturan tersebut para peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) tidak dapat asal dalam mengisi jawaban karena apabila peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) menjawab salah maka nilai akan dikurangi -1. Berikut adalah ilustrasi dari sistem penilaian SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

Tabel 3.4 Tabel Simulasi Penilaian Tes Bidang Studi Prediktif

Soal No	Peserta A		Peserta B	
	Jawaban	Skor	Jawaban	Skor
1	Benar	4	Benar	4
2	Benar	4	Kosong	0
3	Salah	-1	Kosong	0
4	Salah	-1	Kosong	0
5	Salah	-1	Kosong	0
6	Salah	-1	Kosong	0
7	Salah	-1	Kosong	0
Jumlah		3		4

Dari ilustrasi diatas dapat disimpulkan bahwa peserta jangan terlalu bernafsu dan berspekulasi dalam mengisi soal-soal test.

3.1.4 Try Out SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri)

Melihat ketatnya persaingan dalam masuk ke perguruan tinggi negeri, para peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) harus mempersiapkan diri dengan baik. Pengenalan soal-soal dan sistem penilaian SNMPTN menjadi salah satu persiapan dalam menghadapi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri). Sehingga para peserta dapat membuat strategi dalam menghadapi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri). *Try Out* SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) menjadi sangat penting dalam menghadapi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri), agar para peserta dapat mengetahui bagaimana kemampuan dan kesiapan dalam menghadapi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri). Sehingga para peserta dapat mengetahui dan melengkapi apa saja kekurangan yang ada pada peserta.

3.2 Analisis Masalah

pendidikan formal sangat penting bagi setiap orang. Pendidikan yang bersifat sistematis, berstruktur dan bertingkat dimulai dari sekolah dasar sampai keperguruan tinggi. Pendidikan formal merupakan bagian dari pendidikan nasional yang bertujuan untuk membentuk seseorang yang dapat bersaing di era global saat ini.

Dalam pendidikan formal yang terstruktur dan berjenjang terdiri dari pendidikan dasar, pendidikan menengah, dan pendidikan tinggi. Pemilihan tempat menjalani pendidikan sangatlah penting, terlebih dalam

kenaikan jenjang pendidikan dari jenjang SMA/ sederajat ke perguruan tinggi. Sebagian siswa yang baru lulus SMA/ sederajat cenderung memilih perguruan tinggi negeri dibandingkan dengan perguruan tinggi swasta. Hal ini dikarenakan biaya yang lebih terjangkau, selain itu dari segi kualitas pendidikan perguruan tinggi negeri juga lebih baik dengan adanya tenaga pengajar yang handal dan berkompeten dibidangnya, dengan begitu diharapkan lulusan perguruan tinggi negeri mempunyai kualitas dan dapat bersaing di era global saat ini. Namun perguruan tinggi negeri memiliki keterbatasan daya tampung, sehingga perguruan tinggi negeri melakukan SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

Melihat pentingnya SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri), para peserta pun tidak tinggal diam, para peserta yang ingin lolos dalam SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dan masuk ke perguruan tinggi negeri melakukan persiapan, yang salah satunya adalah *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri), sehingga Para peserta dapat mengetahui seberapa besar kemampuannya dalam mengikuti SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri).

Namun *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) yang diadakan di sekolah atau ditempat yang menyediakan bimbingan belajar atau bimbel memiliki beberapa keterbatasan, diantaranya :

- 1) Terdapat kendala dalam proses pembelajaran yang masih bersifat langsung/konvensional.
- 2) Keterbatasan waktu penyampaian materi menyebabkan dibutuhkan sebuah media pembelajaran lain di luar jam sekolah.
- 3) Proses pengerjaan dan pemeriksaan *Try Out* membutuhkan waktu lama, dan penggunaan kertas untuk lembar soal maupun

lembar jawaban sering terjadi kerusakan yang menyebabkan kesalahan dalam pemeriksaan;

- 4) Kurangnya perhatian siswa terhadap materi yang disampaikan di dalam kelas dan kurang banyaknya referensi bahan pembelajaran yang diberikan di kelas.

3.3 Rancangan Solusi Pemecahan

- 1) Membuat sebuah alternatif *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) tanpa harus menggunakan lembaran *fotocopy* atau buku.
- 2) Membuat sebuah *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) yang efektif dan mudah dimengerti.
- 3) Membuat sebuah *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) yang bersifat *mobile* yang dapat memberikan kemudahan dalam melakukan *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri), sehingga *Try Out* atau uji coba SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) dapat dilakukan dimana saja dan kapan saja.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Metode Perancangan

Dalam pengembangan sistem yang akan dibangun, penulis menggunakan pemodelan UML (*Unified Modelling Language*) sedangkan aplikasi yang akan dibangun merupakan sistem aplikasi yang berjalan pada sistem operasi android dan akan bernama *Try Out SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) pada android*. Aplikasi tersebut menyediakan pelatihan atau simulasi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dan dengan sistem *database* MySQL.

4.2 Perancangan Sistem

Pada BAB ini akan membahas bagaimana perancangan sistem yang akan dibangun yang diharapkan dapat membantu para peserta yang ingin menghadapi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri). Perancangan sistem yang akan dibangun mencakup 3 tahap perancangan yaitu :

- 1) Perancangan *Database*.
- 2) Perancangan tampilan aplikasi.
- 3) Perancangan admin untuk meng-Update soal.

Dalam bab ini pengembangan atau perancangan sistem akan dilakukan dengan cara pemodelan menggunakan UML (*Unified Modelling Language*) yang terdiri dari :

- 1) *Use Case diagram*
- 2) *Class diagram*

- 3) *Sequence diagram*
- 4) *Collaboration diagram*
- 5) *Activity diagram*
- 6) *Statechart diagram*
- 7) *Development diagram*

4.2.1 Use Case Diagram

Tahapan yang akan dilakukan dalam membuat *Use Case Diagram* adalah menentukan kandidat *actor*, menentukan *Use Case Requirement* yang berguna untuk mengidentifikasi kebutuhan *usecase* dalam Aplikasi Sistem Informasi Geografis, dan interaksi antara *actor-actor* terhadap *usecase-usecase* yang telah didefinisikan melalui pembuatan *use case diagram*. Berikut penjabarannya:

1. Actor

Actor-actor yang terdapat dalam sistem ini adalah

- *Administrator*: mempunyai hak akses untuk melakukan *input*, *update* dan *delete* ujian, soal, jawaban, nilai, dan user.
- *User* : dapat melakukan tryout SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dan melihat nilai yang didapat dari tryout yang dilakukan oleh user.

2. Use Case Requirement

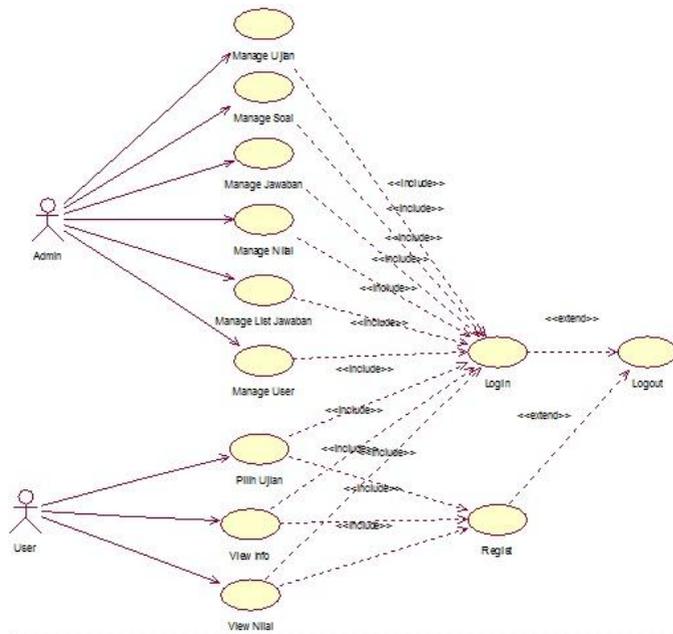
Setelah mengidentifikasi *actor*, tahap selanjutnya adalah menentukan kebutuhan fungsi *usecase* yang dibutuhkan oleh sistem dalam interaksinya dengan *actor-actor* tersebut. Berikut adalah identifikasi kebutuhan *use case*

Aplikasi Sistem triout SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negri) :

Tabel 4.1 : *Use Case Requirement*

No	Requirement	Actor	Usecase
1	Sebelum masuk kedalam sistem, <i>Admin</i> harus <i>login</i> terlebih dahulu	<i>Admin</i>	<i>Login</i>
2	<i>Admin</i> dapat menghapus, mengubah dan menambah Ujian.	<i>Admin</i>	<i>Manage Ujian</i>
3	<i>Admin</i> dapat menambah, menghapus, mengubah data soal.	<i>Admin</i>	<i>Manage Soal</i>
4	<i>Admin</i> dapat menambah, menghapus, mengubah Jawaban.	<i>Admin</i>	<i>Manage Jawaban</i>
5	<i>Admin</i> dapat menambah, menghapus, mengubah nilai.	<i>Admin</i>	<i>Manage Nilai</i>
5	<i>Admin</i> dapat menambah, menghapus, mengubah list Jawaban.	<i>Admin</i>	<i>Manage list Jawaban</i>
5	<i>Admin</i> dapat menambah, menghapus, mengubah User.	<i>Admin</i>	<i>Manage User</i>
6	Sebelum masuk kedalam sistem, User harus <i>login</i> terlebih dahulu	<i>User</i>	<i>Login</i>

7	User memperoleh/ mendapatkan informasi.	User	Info
8	User dapat melakukan Tryout SNMPTN	User	Ujian
9	User memperoleh/mendapatkan informasi Nilai Tryout	User	Nilai

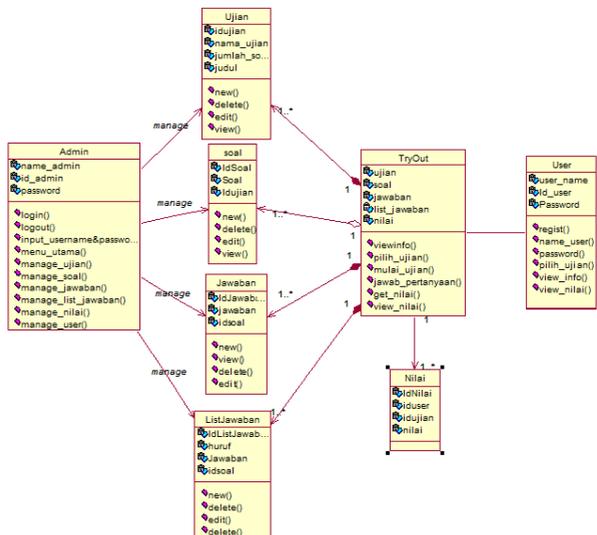


Gambar 4.1 Use Case Diagram

4.2.2 Class diagram

Setelah membuat *use case diagram*, selanjutnya adalah membuat *class diagram*. *Class Diagram* merupakan diagram yang menunjukkan kebutuhan *class-class* dalam suatu sistem program dimana *class-class* tersebut mengandung atribut - atribut dan *operation-operation* yang dibutuhkan. *Class* juga merupakan pembentuk utama dari sistem berorientasi obyek karena *class* menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*.

Dalam satu *Class*, terdapat atribut dan *operation* yang berkaitan dengan *class* tersebut. Kemudian *class-class* yang telah didefinisikan dihubungkan satu sama lain melalui relasi yang terjadi antara *class-class* tersebut.



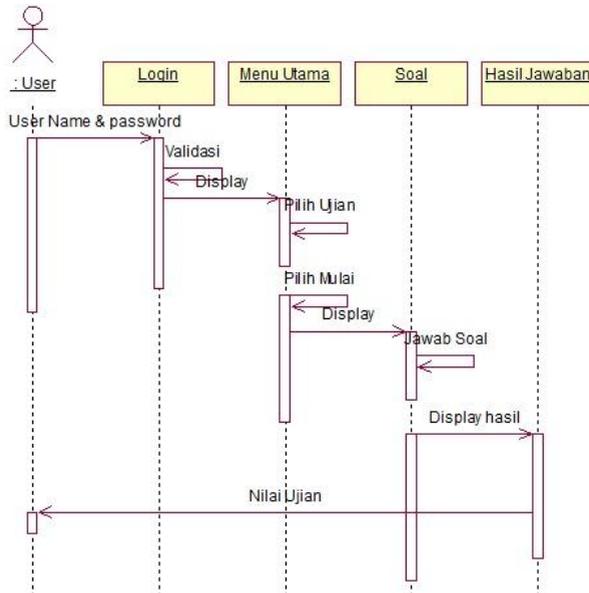
Gambar 4.2 Class Diagram

4.2.3 Sequence Diagram

Sequence Diagram menggambarkan pola hubungan antara sekumpulan *object* yang saling mempengaruhi menurut urutan waktu.

4.2.3.1 Sequence Diagram Try Out (Uji Coba)

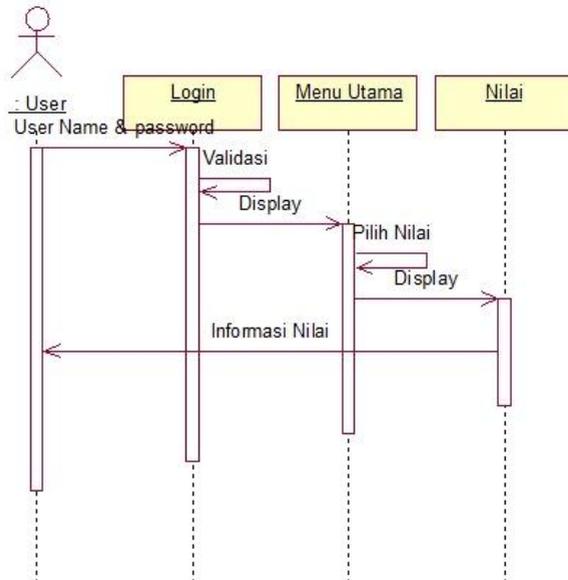
Sebelum masuk kemenu utama *user* diharuskan *login* dengan memasukan *username* dan *password* terlebih dahulu, lalu user memilih ingin melakukan ujian *try out* yang telah di sediakan, setelah memilih ujian *try out* sistem akan memperlihatkan soal-soal yang harus, apabila *user* sudah selesai mengerjakan soal-soal ujian *try out*, sistem akan memberikan pesan berapa nilai yang didapatkan dari mengerjakan soal-soal ujian *try out*.



Gambar 4.3 Sequence Diagram Try Out (Uji Coba)

4.2.3.2 Sequence Diagram View Nilai

User diharuskan melakukan *login* dengan memasukkan *username* dan *password* terlebih dahulu lalu *user* memilih menu nilai pada menu utama, kemudian sistem akan menampilkan daftar nilai.

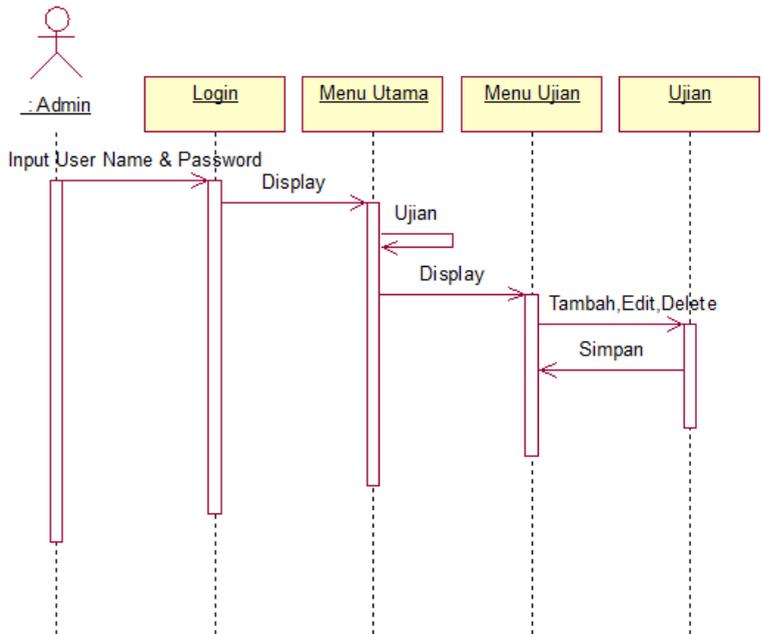


Gambar 4.4 Sequence Diagram View Nilai

4.2.3.3 Sequence Diagram Manage Ujian

Sequence *manage* Ujian melihat proses administrator melakukan *update* Ujian. Untuk masuk kedalam menu utama administrator harus melakukan login dengan memasukkan *user name* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update*

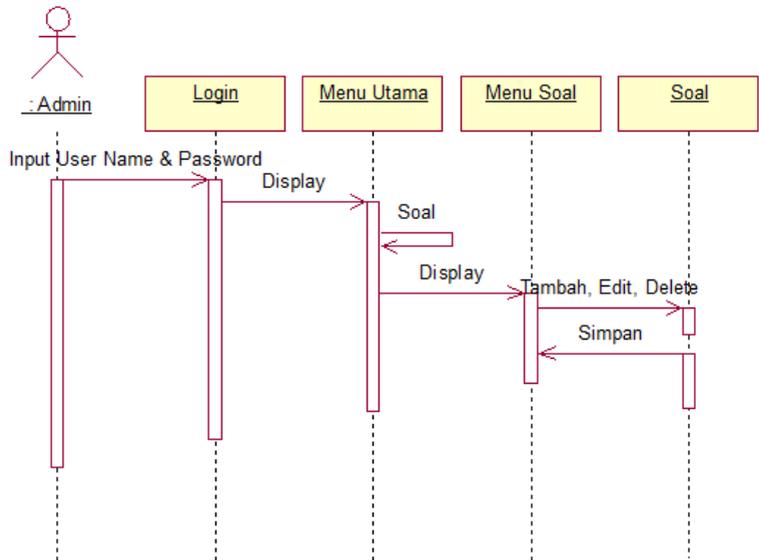
ujian. Setelah selesai *update* ujian maka data tersimpan pada *database* ujian.



Gambar 4.5 *Sequence Diagram Manage Ujian*

4.2.3.4 *Sequence Diagram Manage Soal*

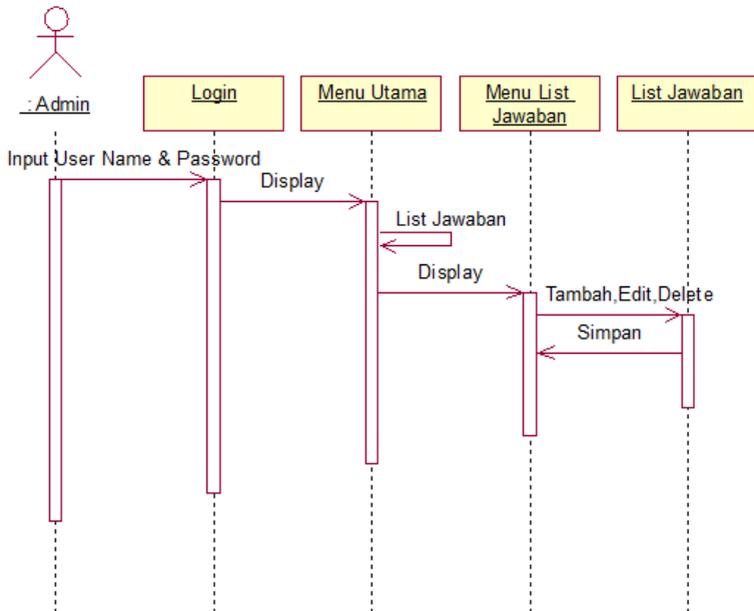
Sequence *manage* soal melihat proses administrator melakukan *update* soal. Untuk masuk kedalam menu utama administrator harus melakukan login dengan memasukkan *user name* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update* soal. Setelah selesai *update* soal maka data tersimpan pada *database* soal.



Gambar 4.6 *Sequence Diagram Manage Soal*

4.2.3.5 *Sequence Diagram Manage Jawaban*

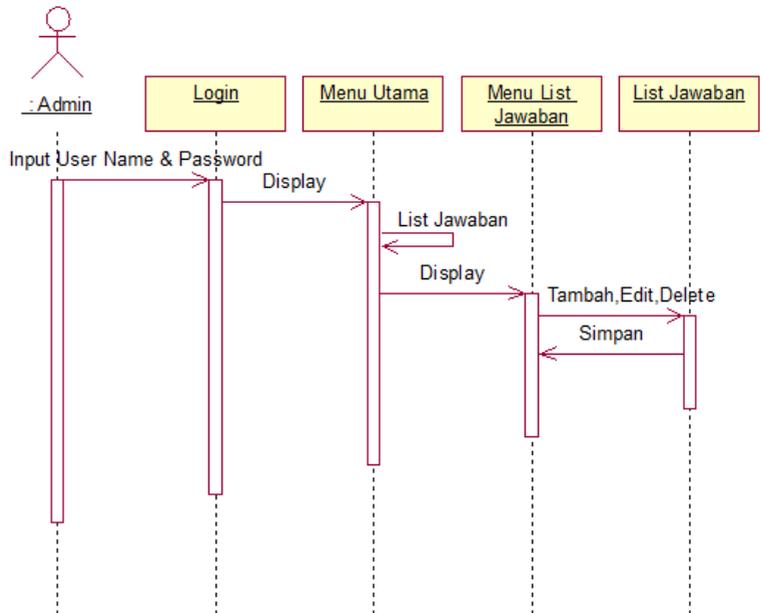
Sequence manage Jawaban melihat proses administrator melakukan *update* Jawaban. Untuk masuk kedalam menu utama administrator harus melakukan login dengan memasukan *user name* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update* Jawaban. Setelah selesai *update* Jawaban maka data tersimpan pada *database* Jawaban.



Gambar 4.7 *Sequence Diagram Manage Jawaban*

4.2.3.6 *Sequence Diagram Manage List Jawaban*

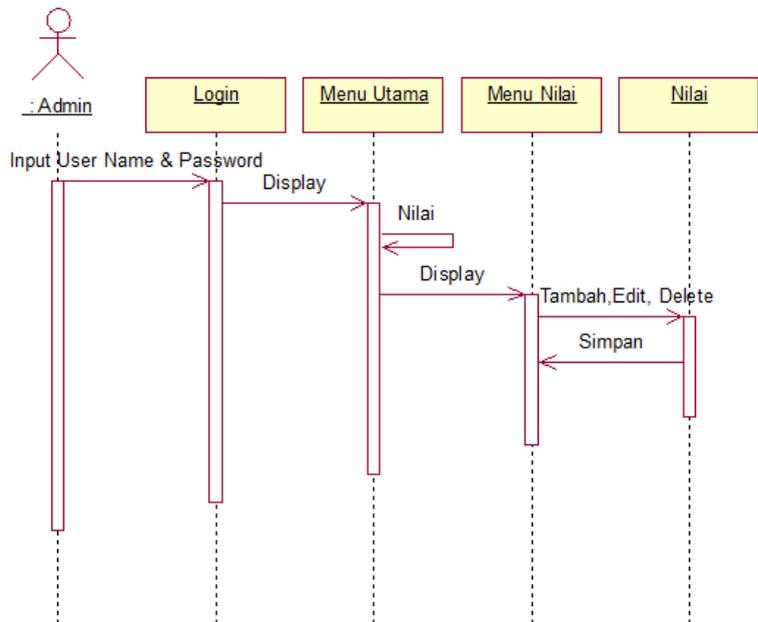
Sequence *manage list Jawaban* melihat proses administrator melakukan *update* list Jawaban. Untuk masuk kedalam menu utama administrator harus melakukan login dengan memasukan *user name* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update* list Jawaban. Setelah selesai *update* list jawaban maka data tersimpan pada *database* list Jawaban.



Gambar 4.8 Sequence Diagram Manage List Jawaban

4.2.3.7 Sequence Diagram Manage Nilai

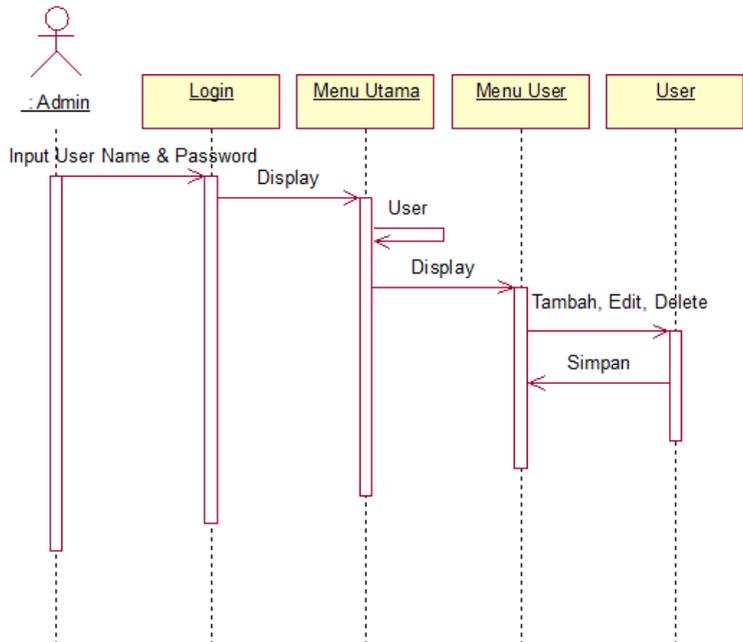
Sequence *manage* nilai melihatkan proses administrator melakukan *update* nilai. Untuk masuk kedalam menu utama administrator harus melakukan login dengan memasukkan *user name* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update* nilai. Setelah selesai *update* nilai maka data tersimpan pada *database* nilai.



Gambar 4.9 Sequence Diagram Manage Nilai

4.2.3.8 Sequence Diagram Manage User

Sequence *manage User* melihatkan proses administrator melakukan *update User*. Untuk masuk kedalam menu utama administrator harus melakukan login dengan memasukkan *user name* dan *password* pada menu *login*, setelah *login* akan terlihat menu utama dan pilih menu *update User*. Setelah selesai *update User* maka data tersimpan pada *database User*.



Gambar 4.10 *Sequence Diagram Manage User*

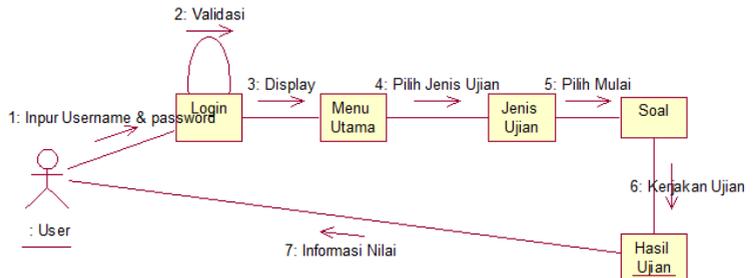
4.2.4 Collaboration Diagram

Collaboration diagram merupakan gambaran lain dari *sequence* diagram yang menunjukkan bagaimana *object* saling terkait satu sama lain. Dalam *collaboration* diagram, digambarkan pengiriman *message* – *message* dari objek satu ke obyek yang lain.

4.2.4.1 *Collaboration Diagram Try Out* (Uji Coba)

User diharuskan melakukan *login* dengan memasukan *username* dan *password* terlebih dahulu lalu *User* memilih ujian pada menu utama dan pilih mulai untuk memulai *try out*

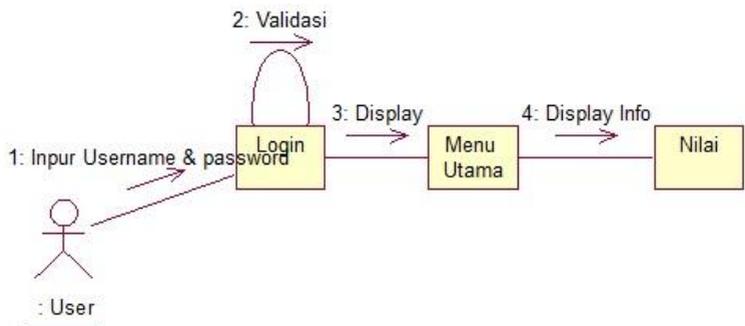
SNMPTN, kemudian sistem akan menampilkan soal-soal Keseluruhan.



Gambar 4.11 *Collaboratin Diagram Try Out*

4.2.4.2 Collaboration Diagram View Nilai

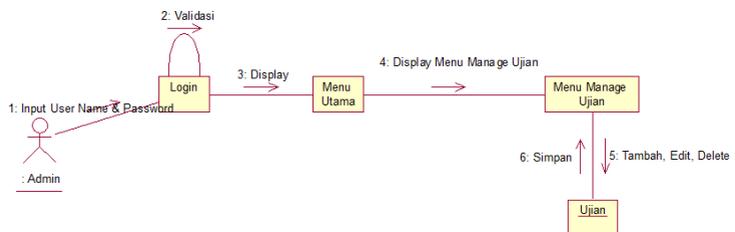
User diharuskan melakukan *login* dengan memasukan *username* dan *password* terlebih dahulu lalu User memilih nilai pada menu utama, kemudian sistem akan menampilkan nilai keseluruhan.



Gambar 4.12 *Collaboratin Diagram Nilai*

4.2.4.3 Collaboration Diagram Manage Ujian

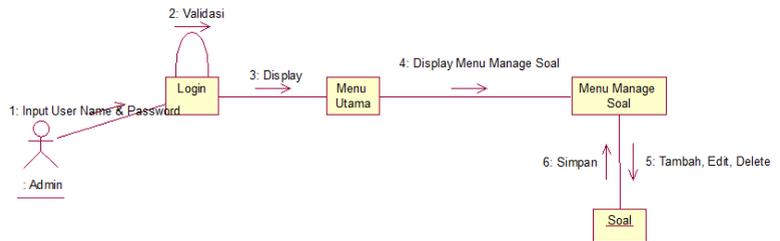
Collaboration Manage ujian memperlihatkan proses *administrator* melakukan *Update* data ujian. Untuk masuk kedalam halaman menu utama, *administrator* harus melakukan *login* terlebih dahulu dengan memasukkan *User Name* dan *Password*. pada menu *login* yang ada setelah *login* akan melihat menu utama dan memilih menu *Update* ujian. Setelah melakukan *Update* data ujian maka data disimpan pada *database* data ujian.



Gambar 4.13 Calloboratin Diagram Manage Ujian

4.2.4.4 Collaboration Diagram Manage Soal

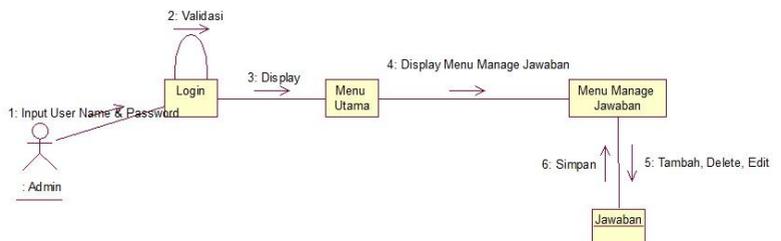
Collaboration Manage soal memperlihatkan proses *administrator* melakukan *Update* data soal. Untuk masuk kedalam halaman menu utama, *administrator* harus melakukan *login* terlebih dahulu dengan memasukkan *User Name* dan *Password*. pada menu *login* yang ada setelah *login* akan melihat menu utama dan memilih menu *Update* soal. Setelah melakukan *Update* data soal maka data disimpan pada *database* data soal.



Gambar 4.14 *Calloboratin* Diagram *Manage Soal*

4.2.4.5 *Collaboration* Diagram *Manage Jawaban*

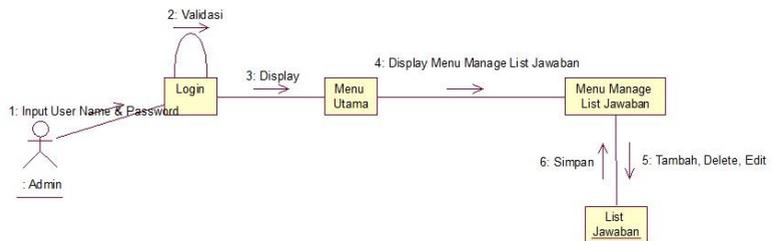
Collaboration Manage jawaban memperlihatkan proses *administrator* melakukan *Update* data jawaban. Untuk masuk kedalam halaman menu utama, *administrator* harus melakukan *login* terlebih dahulu dengan memasukkan *User Name* dan *Password*. pada menu *login* yang ada setelah *login* akan melihat menu utama dan memilih menu *Update* jawaban. Setelah melakukan *Update* data jawaban maka data disimpan pada *database* data jawaban.



Gambar 4.15 *Calloboratin* Diagram *Manage Jawaban*

4.2.4.6 Collaboration Diagram Manage List Jawaban

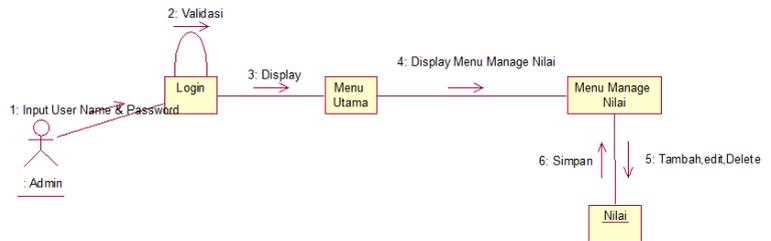
Collaboration Manage list jawaban memperlihatkan proses *administrator* melakukan *Update* data list jawaban. Untuk masuk kedalam halaman menu utama, *administrator* harus melakukan *login* terlebih dahulu dengan memasukkan *User Name* dan *Password*. pada menu *login* yang ada setelah *login* akan melihat menu utama dan memilih menu *Update* list jawaban. Setelah melakukan *Update* data list jawaban maka data disimpan pada *database* data list jawaban.



Gambar 4.16 Collaboration Diagram Manage List Jawaban

4.2.4.7 Collaboration Diagram Manage Nilai

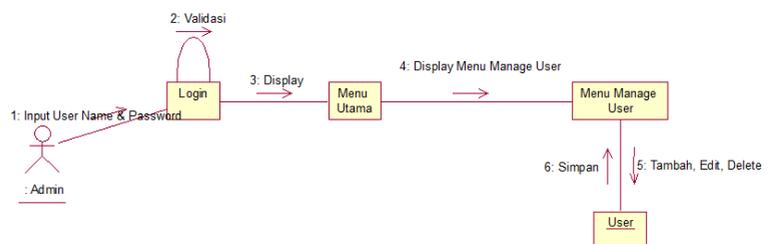
Collaboration Manage nilai memperlihatkan proses *administrator* melakukan *Update* data nilai. Untuk masuk kedalam halaman menu utama, *administrator* harus melakukan *login* terlebih dahulu dengan memasukkan *User Name* dan *Password*. pada menu *login* yang ada setelah *login* akan melihat menu utama dan memilih menu *Update* nilai. Setelah melakukan *Update* data nilai maka data disimpan pada *database* data nilai.



Gambar 4.17 Calloboratin Diagram Manage Nilai

4.2.4.8 Collaboration Diagram Manage User

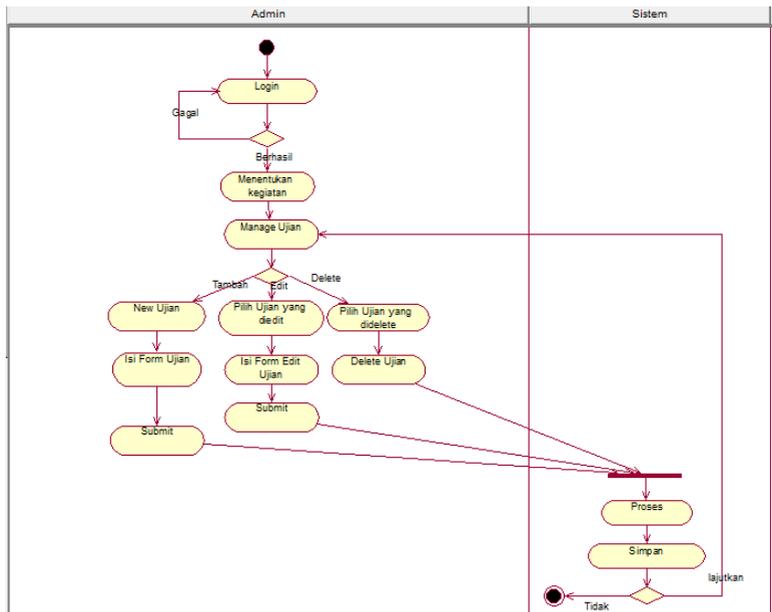
Collaboration Manage User memperlihatkan proses administrator melakukan *Update* data *User*. Untuk masuk kedalam halaman menu utama, administrator harus melakukan *login* terlebih dahulu dengan memasukkan *User Name* dan *Password*. pada menu *login* yang ada setelah *login* akan melihat menu utama dan memilih menu *Update User*. Setelah melakukan *Update* data *User* maka data disimpan pada *database* data *User*.



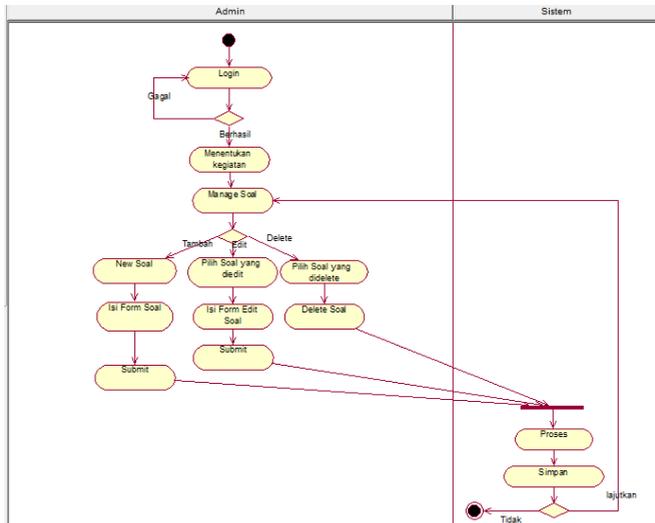
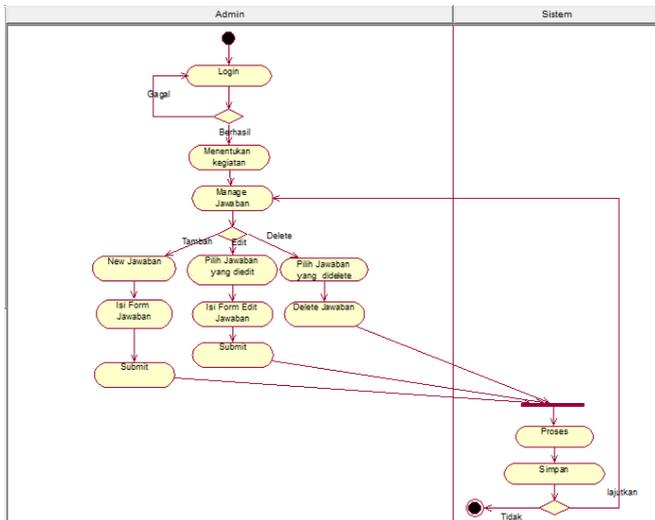
Gambar 4.18 Calloboratin Diagram Manage User

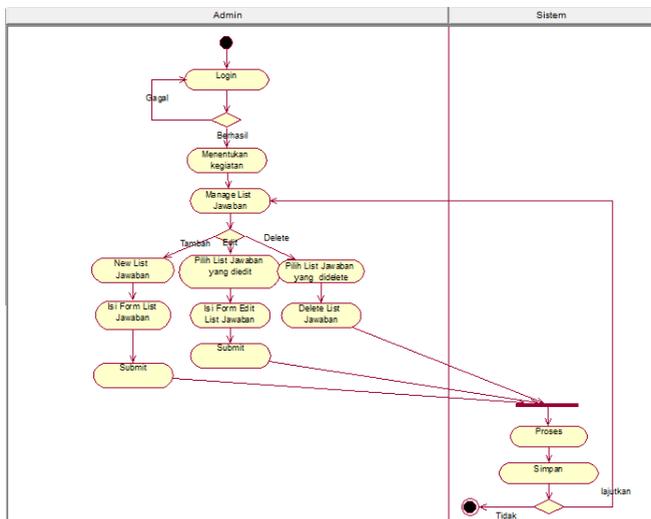
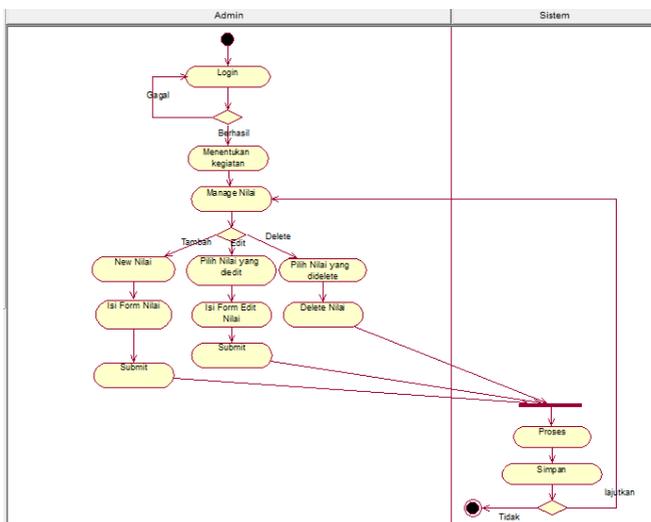
4.2.5 Activity Diagram

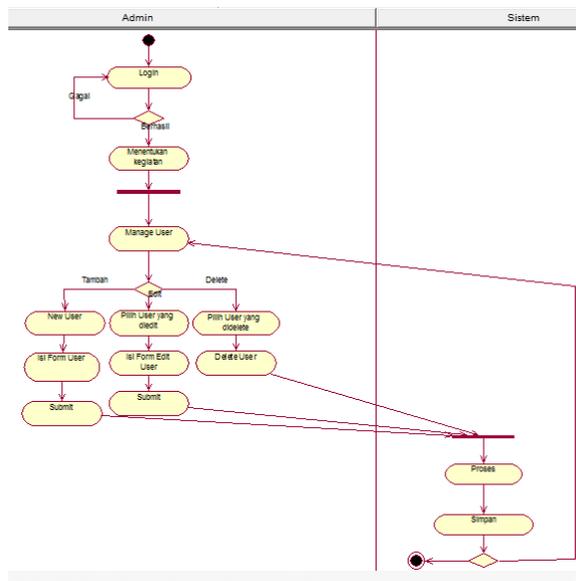
Activity Diagram merupakan gambaran detail dan use case diagram dimana setiap state merupakan suatu aksi (*action state*) dan transisinya dipicu oleh aksi (*action*) yang sudah selesai dan state sebelumnya dan biasanya digunakan untuk menunjukkan urutan dan *state-state*. Berikut adalah *activity diagram* management Try out:



Gambar 4.19 Activity Diagram Manage Ujian

Gambar 4.20 Activity Diagram *Manage Soal*Gambar 4.21 Activity Diagram *Manage Jawaban*

Gambar 4.22 Activity Diagram *Manage List Jawaban*Gambar 4.23 Activity Diagram *Manage Nilai*



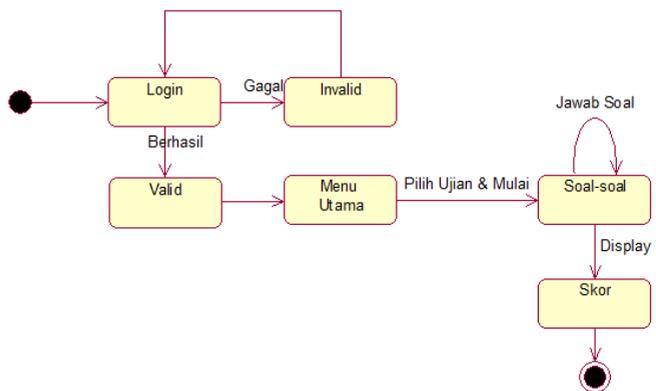
Gambar 4.24 Activity Diagram *Manage User*

4.2.6 Statechart Diagram

Sebuah *statechart* diagram (juga disebut *state* diagram) merupakan bentuk lain dari pemodelan dinamik yang menekankan pada *event-event* yang terjadi pada sebuah *object* sebagai responnya terhadap *message*. *State* diagram menelusuri individu-individu *object* melalui keseluruhan daur hidupnya, menspesifikasikan semua urutan yang mungkin dari pesan-pesan yang akan diterima *object* tersebut, bersama-sama dengan tanggapan atas pesan-pesan tersebut.

4.2.6.1 Statechart Diagram Try Out (Uji Coba)

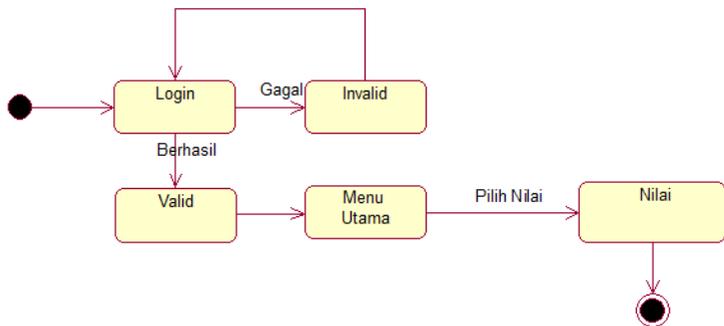
User harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih ujian lalu pilih mulai dan mengerjakan soal-soal, apabila sudah selesai mengerjakan soal-soal sistem akan menampilkan nilai dari ujian yang dilakukan oleh *user*.



Gambar 4.24 Activity Diagram Try Out (Uji Coba)

4.2.6.2 Statechart Diagram View Nilai

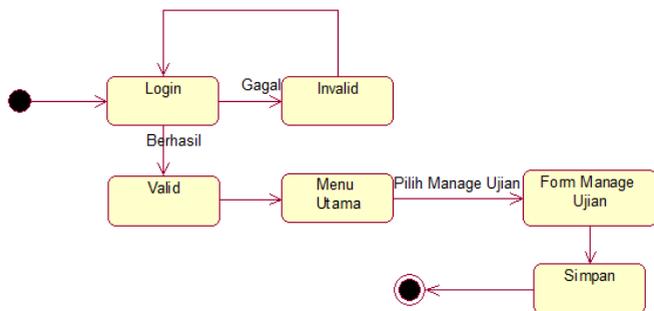
User harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih nilai sistem akan menampilkan nilai dari ujian.



Gambar 4.25 Statechart Diagram View Nilai

4.2.6.3 Statechart Diagram Manage Ujian

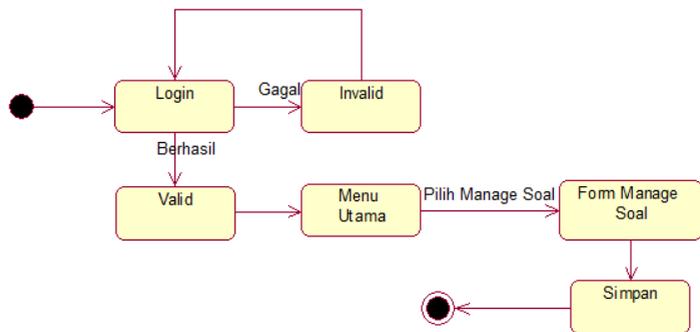
Admin harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih *Manage Ujian* sistem akan menampilkan Form *update* ujian, setelah selesai update lalu simpan.



Gambar 4.26 Statechart Diagram Manage Ujian

4.2.6.4 Statechart Diagram Manage Soal

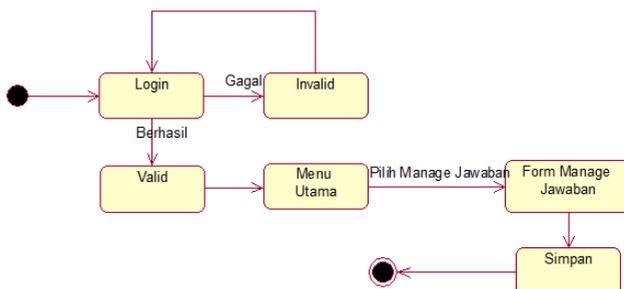
Admin harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih *Manage Soal* sistem akan menampilkan Form *update* soal, setelah selesai update lalu simpan.



Gambar 4.27 Statechart Diagram Manage Ujian

4.2.6.5 Statechart Diagram Manage Jawaban

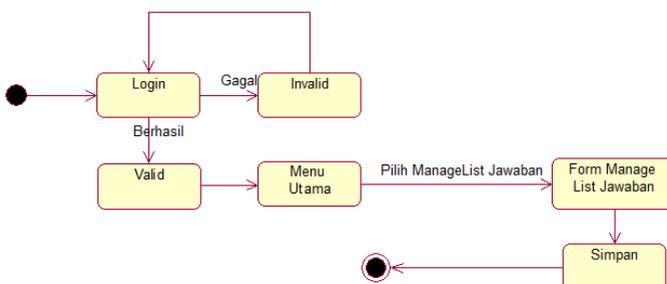
Admin harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih *Manage Jawaban* sistem akan menampilkan Form *update* Jawaban, setelah selesai update lalu simpan.



Gambar 4.28 Statechart Diagram Manage Jawaban

4.2.6.6 Statechart Diagram Manage List Jawaban

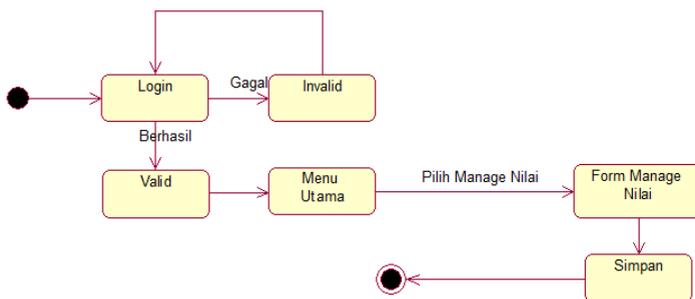
Admin harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih *Manage List Jawaban* sistem akan menampilkan Form *update List Jawaban*, setelah selesai update lalu simpan.



Gambar 4.29 Statechart Diagram Manage List Jawaban

4.2.6.7 Statechart Diagram Manage Nilai

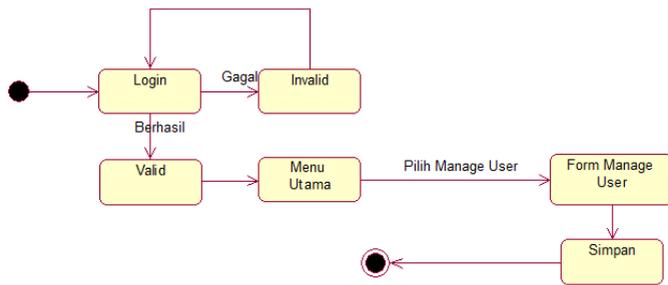
Admin harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih *Manage Nilai* sistem akan menampilkan Form *update* Nilai, setelah selesai update lalu simpan.



Gambar 4.30 Statechart Diagram Manage Nilai

4.2.6.8 Statechart Diagram Manage User

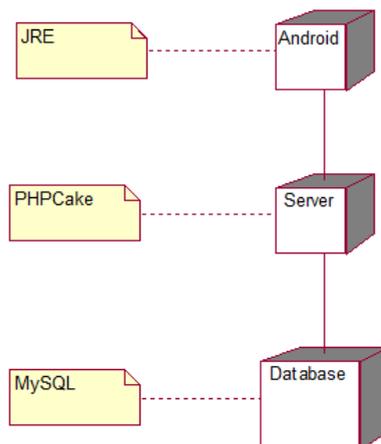
Admin harus melakukan *login* terlebih dahulu dengan memasukkan *user name* dan *password*. pada menu *login* yang ada, setelah *login* akan terlihat menu utama dan memilih *Manage User* sistem akan menampilkan Form *update* User, setelah selesai update lalu simpan.



Gambar 4.31 Statechart Diagram *Manage User*

4.2.7 Deployment Diagram

Deployment diagram menunjukkan sesunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras.



Gambar 4.33 *Deployment Diagram*

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Adapun kesimpulan yang dapat diambil dari Perancangan Sistem Try Out (Uji Coba) SNMPTN adalah sebagai berikut:

1. Telah dibuat sebuah Try Out (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi) *online*, yang menyajikan simulasi ujian SNMPT (Seleksi Nasional Masuk Perguruan Tinggi Negeri).
2. Memberikan gambaran bagaimana kemampuan calon peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) sehingga para calon peserta SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dapat lebih menyiapkan diri.
3. Sistem yang dibangun bersifat dinamis sehingga data-data dapat dilengkapi/*diupdate*.

5.2 Saran

Adapun saran dari penulis setelah merancang dan membangun Try Out (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri), antara lain :

1. Rancangan sistem penunjang keputusan perekrutan dan penempatan karyawan dapat diimplementasikan agar para peserta yang akan menghadapi SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) dapat menyiapkan diri dan dapat bersaing dalam mendapatkan posisi sebagai mahasiswa Perguruan Tinggi Negeri.

2. Penulis berharap, Try Out (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) ini dapat dikembangkan lagi, sehingga Try Out (Uji Coba) tidak hanya untuk SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri)
3. Dalam mengelola database soal Try Out (Uji Coba) SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) administrator menyediakan soal-soal ujian SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri) yang sudah diujikan ditahun sebelumnya.

DAFTAR PUSTAKA

- Byous, Johnson, (2006) Java Technology : The Early Years. California U.S.A: Sun Developer Network.
- Developer Android.2010. 3 Juni 2012, <http://www.developer.android.com>
- Thomas,C & Carolin,B.(2005). *DATA BASE SYSTEMS Apractical Approach to Design, Implementation, and Management.*
- Cocobook, 2010, 1 Juni 2012, <http://book.cakephp.org/id/>.
- Mahmud, Subaena.2009.3 Juni 2012, <http://www.ilmukomputer.com>.
- Munawar. 2005. Pemodelan Visual Dengan UML, Yogyakarta: Graha Ilmu.
- Safaat, Nazarudin.(2011).Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Bandung: Informatika.
- Setiawan, Yudha.(2004).Database Client Server Menggunakan Internet Base, Yogyakarta: ANDI.

IDENTITAS PENYUSUN**DATA PRIBADI**

NIM : 2007-81-049
Nama : Kristian Ariyanto Zai
Jurusan : Teknik Informatika
TTL : Tangerang, 20 Oktober 1988
Agama : Kristen
Jenis Kelamin : Laki-Laki
Alamat : Jl.Borobudur A67 No.24 RT.05 RW.01, Pinang
Indah Kota Tngerang, Banten

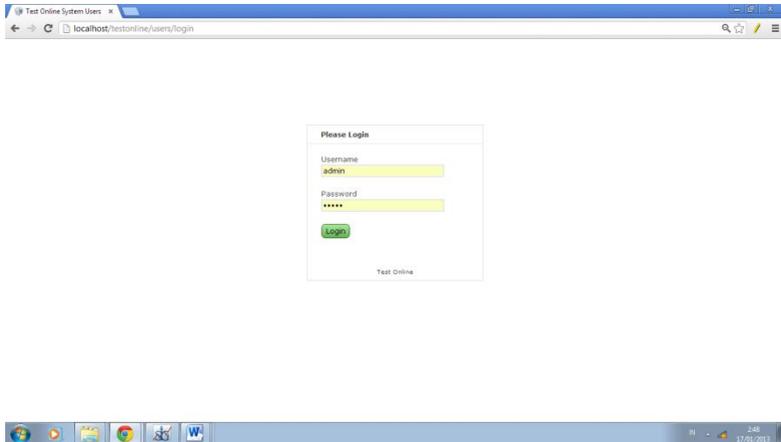
PENDIDIKAN

Tahun 1995-2001 : SDN Bojong 2, Tangerang
Tahun 2001-2004 : SMP Yadika 3, Ciledug
Tahun 2004-2007 : SMA Yadika 3, Ciledug
Tahun 2007-2012 : Universitas Esa Unggul

LAMPIRAN

A. Admin

Form Login



Home



New Mata Pelajaran

Test Online System Mata | x
localhost/testonline/mata_pelajarans/add

17 Jan 2013
User: admin

My Panel Command Center Preferences Logout

Actions

- List Mata Pelajarans
- List Nilai
- New Nilai
- List Soals
- New Soal

Add Mata Pelajaran

Nama

Jumlah Soal

Judul

Submit

Online Test Application
(Default) 15 questions took 111 ms

Form New Mata Pelajaran

Test Online System Mata | x
localhost/testonline/mata_pelajarans

17 Jan 2013
User: admin

My Panel Command Center Preferences Logout

Actions

- New Mata Pelajaran
- List Nilai
- New Nilai
- List Soals
- New Soal

Mata Pelajaran

Id	Nama	Jumlah Soal	Judul	Actions
1	Matematika	30		View Edit Delete
2	Bahasa Indonesia	30	Pilihah Persamaan Kata Yang Paling Mendekati Dari Soal Obawah Ini	View Edit Delete
6	Bahasa Inggris	30	Isilah titik-titik berikut dengan tepat	View Edit Delete
7	Test Pengetahuan Umum	5	Bahasa	View Edit Delete

Page 1 of 1, showing 4 records out of 4 total, starting on record 1, ending on 4
[← previous](#) | [next →](#)

Online Test Application
(Default) 17 questions took 120 ms

New Soal

Test Online System Soals | x
localhost/testonline/soals

17 Jan 2013
User: admin

My Panel Command Center Preferences Logout

Actions

- New Soal
- List Mata Pelajarans
- New Mata Pelajaran
- List Jawabans
- New Jawabans
- List List Jawabans
- New List Jawabans

Soal

Id	Soal	Mata Pelajaran	Actions
1	MILITAN	Bahasa Indonesia	View Edit Delete
2	EKSKAVASI	Bahasa Indonesia	View Edit Delete
3	PARADOKSAL	Bahasa Indonesia	View Edit Delete
4	Pembatasan	Bahasa Indonesia	View Edit Delete
5	Harta benda	Bahasa Indonesia	View Edit Delete
6	Ciri	Bahasa Indonesia	View Edit Delete
7	EVOKASI	Bahasa Indonesia	View Edit Delete
8	DAUR	Bahasa Indonesia	View Edit Delete
9	SINTEGIS	Bahasa Indonesia	View Edit Delete

Online Test Application
(Default) 17 questions took 120 ms

Form New Soal

Test Online System Soals

localhost/testonline/soals/add

17 Jan 2013
User: admin

TE T
online

My Panel | Command Center | Preferences | Logout

Actions

- List Soals
- List Mata Pelajaran
- New Mata Pelajaran
- List Jawabans
- New Jawaban
- List List Jawabans
- New List Jawaban

Add Soal

Soal

Mata Pelajaran
Matematika

Submit

Online Test Application
(default) 10 questions took 60 ms

New Jawaban

Test Online System Jawabans

localhost/testonline/jawabans

17 Jan 2013
User: admin

TE T
online

My Panel | Command Center | Preferences | Logout

Actions

- New Jawaban
- List Soals
- New Soal

Jawaban

Id	Soal	Jawaban	Actions
1	MILITAN	c	View Edit Delete
2	EKSKAVASI	a	View Edit Delete
3	PARADOKSAL	d	View Edit Delete
4	Pembatasan	d	View Edit Delete
5	Harta benda	a	View Edit Delete
6	Ciri	c	View Edit Delete
7	EYOKASI	d	View Edit Delete
8	DAUR	a	View Edit Delete
9	SINTEISIS	e	View Edit Delete
10	Deskriptif	c	View Edit Delete
11	Gambaran	e	View Edit Delete

Form New Jawaban

Test Online System Jawabans

localhost/testonline/jawabans

17 Jan 2013
User: admin

TE T
online

My Panel | Command Center | Preferences | Logout

Actions

- New Jawaban
- List Soals
- New Soal

Jawaban

Id	Soal	Jawaban	Actions
1	MILITAN	c	View Edit Delete
2	EKSKAVASI	a	View Edit Delete
3	PARADOKSAL	d	View Edit Delete
4	Pembatasan	d	View Edit Delete
5	Harta benda	a	View Edit Delete
6	Ciri	c	View Edit Delete
7	EYOKASI	d	View Edit Delete
8	DAUR	a	View Edit Delete
9	SINTEISIS	e	View Edit Delete
10	Deskriptif	c	View Edit Delete
11	Gambaran	e	View Edit Delete

Nilai

Test Online System Nilas

localhost/testonline/nilas

17 Jan 2013
User: admin

My Panel Command Center Preferences Logout

Nilai

Id	User	Mata Pelajaran	Skor	Actions
1	Istiraful	Matematika	90	View Edit Delete
2	Istiraful	Bahasa Indonesia	-2	View Edit Delete

Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

[← previous](#) | [next →](#)

Online Test Application
(default) 17 queries took 100 ms

List Jawaban

Test Online System List Ja

localhost/testonline/list_jawabans

17 Jan 2013
User: admin

My Panel Command Center Preferences Logout

List Jawaban

Id	Huruf	Jawaban	Soal	Actions
1	a		Anti pencemaran	View Edit Delete
2	b		Ideologi	View Edit Delete
3	c		Agresif	View Edit Delete
4	d		Brutal	View Edit Delete
5	e		Penggolongan	View Edit Delete
6	a		Penggalian	View Edit Delete
7	b		Pertolongan	View Edit Delete
8	c		Penggangkutan	View Edit Delete
9	d		Pelaku	View Edit Delete

Actions

- [New List Jawaban](#)
- [List Soals](#)
- [New Soal](#)

Online Test Application
(default) 16 queries took 72 ms

New List Jawaban

Test Online System List Ja

localhost/testonline/list_jawabans/add

17 Jan 2013
User: admin

My Panel Command Center Preferences Logout

Add List Jawaban

Huruf

Jawaban

Soal
MILTAN

[Submit](#)

Actions

- [List List Jawabans](#)
- [List Soals](#)
- [New Soal](#)

Online Test Application
(default) 16 queries took 72 ms

User

Test Online System Users | localhost / localhost / app | 17 Jan 2013 | User: admin

User deleted

Actions

- New User
- List Roles
- New Role
- List Nilais
- New Nilai

Users

Id	Username	Fullname	Password	Role	Actions
3	admin	admin	86dc988a6216339b92d72e201e1a5c1404c34ae	Administrator	View Edit Delete
7	zai	tian	827ccb0eea9a706c4c34a16891f84e7b		View Edit Delete

Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2
 << previous | next >>

Online Test Application
 (default) 17 January 2013 1:03 am

B. USER

Form Login

TestOnline

Selamat Datang..
Silahkan Login

Username

Password

Login Reset Register

Form Registrasi



The screenshot shows a mobile application interface for registration. At the top, the status bar displays '3G', signal strength, and the time '3:14 AM'. The app title 'Registrasi' is visible in the header. The main content area has a light gray background with a grid pattern. It contains the text 'Registrasi' and 'Silahkan Registrasi !'. Below this are three input fields: 'Masukkan username' (highlighted with an orange border), 'Masukkan Full Name', and 'Masukkan Password'. At the bottom, there are 'Submit' and 'Reset' buttons, and a link that says 'Already registred. Login Me!'.

Login Sukses



The screenshot shows a mobile application interface for login. The status bar at the top shows '3G', signal strength, and the time '3:16 AM'. The app title 'TestOnline' is in the header. The main content area has a dark gray background with a grid pattern. It displays 'Selamat Datang..' and 'Silahkan Login'. A dialog box is overlaid on the screen, titled 'Password' with a dropdown arrow icon. The dialog box contains the text 'succeeded' and has 'OK' and 'Cancel' buttons at the bottom.

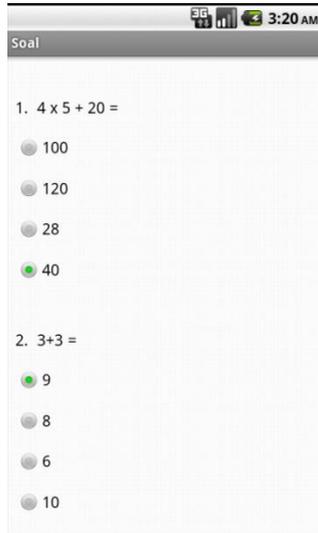
Home



Pilih Mata Pelajaran



Mulai Ujian



The image shows a mobile application interface for a quiz. At the top, there is a status bar with icons for signal strength, Wi-Fi, and battery, along with the time 3:20 AM. Below the status bar is a header labeled "Soal". The main content area contains two math problems, each with four radio button options. The first problem is "1. $4 \times 5 + 20 =$ " with options 100, 120, 28, and 40. The second problem is "2. $3+3 =$ " with options 9, 8, 6, and 10. In both cases, the first option is selected, indicated by a green dot in the center of the radio button.

Soal

1. $4 \times 5 + 20 =$

100

120

28

40

2. $3+3 =$

9

8

6

10