

**APLIKASI *ONLINE JUDGE* DAN *TRACKING* PELAJAR  
BERBASIS KOMPETISI PEMROGRAMAN**

**TUGAS AKHIR**

Oleh:

**JEFFRY TANDIONO**

NIM. 121110812

**MICHAEL**

NIM. 121110804



**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
MIKROSKIL  
MEDAN  
2016**

**ONLINE JUDGE AND STUDENT'S TRACKING  
APPLICATION BASED PROGRAMMING CONTEST**

**FINAL RESEARCH**

By:

**JEFFRY TANDIONO**

ID. 121110812

**MICHAEL**

ID. 121110804



**STUDY PROGRAM OF INFORMATICS ENGINEERING  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
MIKROSKIL  
MEDAN  
2016**

**LEMBARAN PENGESAHAN**

**APLIKASI ONLINE JUDGE DAN TRACKING PELAJAR  
BERBASIS KOMPETISI PEMROGRAMAN**

**TUGAS AKHIR**

Diajukan untuk Melengkapi Persyaratan Guna  
Mendapatkan Gelar Sarjana Strata Satu  
Program Studi Teknik Informatika

Oleh:

**JEFFRY TANDIONO**  
NIM. 121110812

**MICHAEL**  
NIM. 121110804

Disetujui Oleh:

Dosen Pembimbing I,

Hardy, S.Kom., M.Sc.

Dosen Pembimbing II,

Arwin Halim, S.Kom., M.Kom.

Medan, 13 Agustus 2016  
Diketahui dan Disahkan Oleh:



## ABSTRAK

Kompetisi Pemrograman merupakan kontes yang dilakukan untuk menguji kemampuan pemrograman peserta. *Online Judge* di perkuliahan menjadi sebuah kebutuhan yang krusial untuk membantu kemampuan pemrograman mahasiswa. Permasalahan mahasiswa yang akan dipertandingkan pada kompetisi pemrograman muncul karena kurangnya informasi mengenai mahasiswa yang berprestasi.

Metodologi *waterfall* digunakan dalam menyelesaikan aplikasi *Online Judge* yang dapat dipergunakan di perkuliahan. Algoritma ELO *rating* yang didasarkan pada permainan catur digunakan untuk *tracking* pelajar. ELO *rating* mampu memberikan solusi yang optimal dalam mendapatkan pelajar berprestasi oleh karena sistem *rating* yang disediakan telah dipergunakan dengan luas.

Aplikasi *Online Judge* yang dihasilkan dapat diterapkan pada perkuliahan dan mampu mengurutkan mahasiswa yang berprestasi. Mahasiswa atau pelajar tidak hanya diperbolehkan menyelesaikan soal, tetapi mampu membuat soal sehingga bank soal pada *Online Judge* tidak terbatas dari *developer*. Aplikasi *Online Judge* didesain sedemikian rupa sehingga cukup interaktif terhadap pelajar dan memudahkan penggunaan.

Kata kunci : *Online Judge, Tracking, Rating*.

## **KATA PENGANTAR**

Ucapan syukur kepada Tuhan Yang Maha Esa, berkat rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul, Aplikasi Online Judge dan Tracking Pelajar berbasis Kompetisi Pemrograman, sesuai dengan yang direncanakan. Untuk itu puji syukur penulis panjatkan kepada-Nya.

Dalam penyusunan dan penulisan Tugas Akhir ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini, penulis turut menyampaikan ucapan terima kasih kepada:

1. Bapak Hardy, S.Kom., M.Sc., selaku Ketua Program Studi Teknik Informatika dan Pembimbing I yang telah membimbing penulis selama mengerjakan Tugas Akhir ini.
2. Bapak Arwin Halim, S.Kom., M.Kom., selaku Pembimbing II yang telah membimbing penulis selama mengerjakan Tugas Akhir ini.
3. Bapak Dr. Mimpin Ginting, M.S., selaku Ketua STMIK Mikroskil Medan.
4. Bapak Djoni, S.Kom., M.T.I., selaku Wakil Ketua I STMIK Mikroskil Medan.
5. Bapak dan Ibu Dosen yang telah mendidik dan membimbing penulis dalam mengerjakan Tugas Akhir ini.
6. Kedua orang tua kami atas jasa-jasanya, doa dan tidak pernah lelah dalam mendidik penulis sejak kecil.
7. Saudara-saudara kami yang telah memberikan dukungan, semangat, kasih sayang dan bantuan baik secara moril maupun materiil demi lancarnya penyusunan Tugas Akhir ini.
8. Orang yang kami sayangi dari awal sampai sekarang ini telah memberikan motivasi kepada kami.
9. Semua pihak yang telah membantu penulis untuk terselesaiannya Tugas Akhir ini.

Tugas Akhir ini dibuat untuk melengkapi persyaratan guna memperoleh gelar Sarjana Strata Satu pada Program Studi Teknik Informatika, STMIK Mikroskil Medan. Semoga hasil dari Tugas Akhir ini ada manfaatnya bagi pihak yang berkepentingan.

Medan, 13 Agustus 2016

Penulis

## DAFTAR ISI

Abstrak.....	i
Kata Pengantar .....	ii
Daftar Isi .....	iv
Daftar Gambar .....	vi
Daftar Tabel .....	x
Daftar Lampiran.....	xi
Bab I. Pendahuluan.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan dan Manfaat.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Pengembangan Perangkat Lunak.....	3
Bab II. Tinjauan Pustaka.....	5
2.1 Pemrograman.....	5
2.2 Kompetisi Pemrograman .....	7
2.3 Online Judge .....	8
2.3.1 UVa Online Judge .....	8
2.3.2 Jutge.org .....	10
2.3.3 Mooshak .....	11
2.4 Standar Penjurian .....	12
2.5 Tracking Pelajar .....	13
2.6 Elo Rating.....	13
Bab III. Analisis dan Perancangan .....	22
3.1 Analisis .....	22
3.1.1 Analisis Kebutuhan .....	22
3.1.1.1 Kebutuhan Fungsional .....	22
3.1.1.2 Kebutuhan Non Fungsional .....	29
3.1.2 Analisis Proses .....	30
3.2 Perancangan.....	47

3.2.1	Perancangan Sistem Usulan .....	48
3.2.2	Perancangan Tampilan .....	50
3.2.3	Perancangan Basis Data .....	83
Bab IV.	Hasil dan Implementasi.....	87
4.1	Hasil.....	87
4.2	Implementasi .....	113
Bab V.	Kesimpulan dan Saran .....	120
5.1	Kesimpulan.....	120
5.2	Saran.....	120
Daftar Pustaka.....		121

## DAFTAR GAMBAR

Gambar 2.1	Rancangan sistem Uva Online Judge .....	9
Gambar 2.2	Infrastruktur Jutge.org.....	10
Gambar 2.3	Infrastruktur Mooshak .....	11
Gambar 2.4	Rancangan tampilan input.....	17
Gambar 3.1	Interaksi Pengguna dengan Sistem dalam <i>Use Case Diagram</i> ....	23
Gambar 3.2	Flowchart <i>Binary Search rating</i> .....	33
Gambar 3.3	Flowchart <i>getRatingtoRank</i> .....	34
Gambar 3.4	Flowchart <i>getSeed</i> .....	35
Gambar 3.5	Activity Diagram Melaksanakan Kontes .....	48
Gambar 3.6	Activity Diagram Verifikasi <i>User</i> .....	49
Gambar 3.7	Activity Diagram Menyelesaikan Soal .....	49
Gambar 3.8	Activity Diagram <i>Tracking Pelajar</i> .....	49
Gambar 3.9	Perancangan Tampilan Halaman Utama .....	50
Gambar 3.10	Perancangan Tampilan Halaman Login .....	51
Gambar 3.11	Perancangan Tampilan Halama Register .....	52
Gambar 3.12	Perancangan Tampilan Halaman Ranklists User .....	53
Gambar 3.13	Perancangan Tampilan Halaman Ranklists Universitas .....	54
Gambar 3.14	Perancangan Tampilan Halaman Profil .....	55
Gambar 3.15	Perancangan Tampilan Halaman Submissions .....	56
Gambar 3.16	Perancangan Tampilan Halaman Problem Sets .....	57
Gambar 3.17	Perancangan Tampilan Halaman Solve Problem .....	58
Gambar 3.18	Perancangan Tampilan Halaman Create Problem .....	59
Gambar 3.19	Perancangan Tampilan Halaman Contest Member .....	60
Gambar 3.20	Perancangan Tampilan Halaman Dashboard Admin .....	61
Gambar 3.21	Perancangan Tampilan Halaman Universities Admin.....	62
Gambar 3.22	Perancangan Tampilan Halaman Contest Admin .....	63
Gambar 3.23	Perancangan Tampilan Halaman Create Contest Admin .....	64
Gambar 3.24	Perancangan Tampilan Halaman Edit Contest Admin .....	65
Gambar 3.25	Perancangan Tampilan Halaman Problem Contest .....	66

Gambar 3.26	Perancangan Tampilan Halaman Scoreboard Contest.....	67
Gambar 3.27	Perancangan Tampilan Halaman Submissions Contest .....	68
Gambar 3.28	Perancangan Tampilan Halaman Submission Detail Contest .....	69
Gambar 3.29	Perancangan Tampilan Halaman Problem Category .....	70
Gambar 3.30	Perancangan Tampilan Halaman Problem Sets Admin .....	71
Gambar 3.31	Perancangan Tampilan Mobile Live Contest.....	72
Gambar 3.32	Perancangan Tampilan Mobile Scoreboard .....	73
Gambar 3.33	Perancangan Tampilan Mobile Scoreboard Detail .....	74
Gambar 3.34	Perancangan Tampilan Mobile Ranklist.....	75
Gambar 3.35	Perancangan Tampilan Mobile Upcoming Contest .....	76
Gambar 3.36	Perancangan Tampilan Mobile Upcoming Contest Detail .....	77
Gambar 3.37	Perancangan Tampilan Mobile Login.....	78
Gambar 3.38	Perancangan Tampilan Mobile Register.....	79
Gambar 3.39	Perancangan Tampilan Mobile Account.....	80
Gambar 3.40	Perancangan Tampilan Mobile Options.....	81
Gambar 3.41	Perancangan Tampilan Mobile Edit Profile.....	82
Gambar 3.42	Perancangan Tampilan Mobile Change Password.....	83
Gambar 3.43	Class Diagram User Package Basis Data.....	84
Gambar 3.44	Class Diagram Problem Package Basis Data.....	85
Gambar 3.45	Class Diagram Online Judge Package Basis Data .....	86
Gambar 4.1	Tampilan Halaman Depan Aplikasi.....	87
Gambar 4.2	Tampilan Halaman Login .....	88
Gambar 4.3	Tampilan Halaman Register .....	88
Gambar 4.4	Tampilan Ranklists User .....	89
Gambar 4.5	Tampilan Ranklists Universitas .....	89
Gambar 4.6	Tampilan Profil Member.....	90
Gambar 4.7	Tampilan Submission Member .....	90
Gambar 4.8	Tampilan List Problem Sets Member .....	91
Gambar 4.9	Tampilan Solve Problem Member .....	92
Gambar 4.10	Tampilan Create Problem .....	93
Gambar 4.11	Tampilan Daftar Contest Member .....	93

Gambar 4.12	Tampilan Dashboard Admin.....	94
Gambar 4.13	Tampilan Universitas Admin.....	94
Gambar 4.14	Tampilan Daftar Contest Admin.....	95
Gambar 4.15	Tampilan Create Contest.....	95
Gambar 4.16	Tampilan Problem Contest .....	96
Gambar 4.17	Tampilan Scoreboard Contest Running .....	96
Gambar 4.18	Tampilan Scoreboard Contest Freeze .....	97
Gambar 4.19	Tampilan Scoreboard Contest.....	97
Gambar 4.20	Tampilan Submission Contest .....	98
Gambar 4.21	Tampilan Detail Submission Contest.....	99
Gambar 4.22	Tampilan Problem Category .....	100
Gambar 4.23	Tampilan Problem Sets Admin.....	100
Gambar 4.24	Tampilan Mobile Live Contest .....	101
Gambar 4.25	Tampilan Mobile Scoreboard .....	102
Gambar 4.26	Tampilan Mobile Scoreboard Detail.....	103
Gambar 4.27	Tampilan Mobile Ranklist .....	104
Gambar 4.28	Tampilan Mobile Upcoming Contest.....	105
Gambar 4.29	Tampilan Mobile Upcoming Contest Detail.....	106
Gambar 4.30	Tampilan Mobile Login .....	107
Gambar 4.31	Tampilan Mobile Register .....	108
Gambar 4.32	Tampilan Mobile Account .....	109
Gambar 4.33	Tampilan Mobile Options .....	110
Gambar 4.34	Tampilan Mobile Edit Profile .....	111
Gambar 4.35	Tampilan Mobile Change Password .....	112
Gambar 4.36	Hasil Kuesioner Pernyataan 1 .....	114
Gambar 4.37	Hasil Kuesioner Pernyataan 2 .....	114
Gambar 4.38	Hasil Kuesioner Pernyataan 3 .....	115
Gambar 4.39	Hasil Kuesioner Pernyataan 4 .....	115
Gambar 4.40	Hasil Kuesioner Pernyataan 5 .....	116
Gambar 4.41	Hasil Kuesioner Pernyataan 6 .....	116
Gambar 4.42	Hasil Kuesioner Pernyataan 7 .....	117

Gambar 4.43	Hasil Kuesioner Pernyataan 8.....	117
Gambar 4.44	Hasil Kuesioner Pernyataan 9.....	118
Gambar 4.45	Hasil Kuesioner Pernyataan 10.....	118
Gambar 4.46	Hasil Kuesioner Pernyataan 11.....	119

## DAFTAR TABEL

Tabel 2.1	Probabilitas kemenangan terhadap <i>rating</i> .....	18
Tabel 3.1	Deskripsi Aktor .....	24
Tabel 3.2	Tabel Use Case Register .....	24
Tabel 3.3	Tabel Use Case View Ranklists .....	24
Tabel 3.4	Tabel Use Case View Profile Member .....	25
Tabel 3.5	Tabel Use Case Verify User .....	25
Tabel 3.6	Tabel Use Case Approve Contest .....	26
Tabel 3.7	Tabel Use Case Create Contest .....	27
Tabel 3.8	Tabel Use Case Create Problem .....	27
Tabel 3.9	Tabel Use Case Join Contest .....	28
Tabel 3.10	Tabel Use Case Solve Problem.....	29
Tabel 3.11	Tabel Scoreboard .....	30
Tabel 3.12	Tabel Probabilitas kontestan .....	32
Tabel 3.13	Tabel <i>Seed</i> .....	32
Tabel 3.14	Tabel Delta Kontestan.....	44
Tabel 3.15	Tabel Delta dengan <i>Increment</i> Rata-rata .....	45
Tabel 3.16	Tabel Kalkulasi Delta .....	47
Tabel 3.17	Tabel New Rating .....	47
Tabel 4.1	Spesifikasi <i>Server</i> .....	113
Tabel 4.2	Daftar Kontes .....	113

## **DAFTAR LAMPIRAN**

- Lampiran 1. Kuesioner
- Lampiran 2. Sampel Kuesioner 1
- Lampiran 3. Sampel Kuesioner 2

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan kemajuan teknologi dan makin meningkatnya kebutuhan manusia terhadap komputer, kebutuhan terhadap kemampuan pemrograman semakin meningkat. Untuk membantu meningkatkan kebutuhan pemrograman, telah banyak penjurian *online* yang diciptakan. Penjurian *Online (Online Judge)* adalah suatu sistem *online* yang dipergunakan untuk melakukan pengujian terhadap program pada suatu kompetisi pemrograman (Skeina dan Revilla, 2003).

*Online Judge* yang telah ada sekarang ini dapat membantu meningkatkan kemampuan pemrograman pada sisi kontes pemrograman. *Online Judge* juga menyediakan berbagai soal algoritma sebagai sarana untuk melatih kemampuan logika berpikir dan algoritma pemrograman, akan tetapi kurangnya bank soal pada *online judge* menyebabkan pembelajaran pemrograman cukup terbatas. Sistem *Online Judge* yang ada tidak menawarkan pelatihan yang bisa dipakai untuk diimplementasikan di kampus terutama pada *tracking* pelajar.

*Tracking* pelajar digunakan pada perkuliahan untuk melihat tingkat kemampuan pelajar dalam pemrograman dan membandingkan para pelajar untuk mendapatkan pelajar yang berbakat. *Tracking* pelajar berupa *history* perkembangan pelajar dari waktu ke waktu baik dalam simulasi kontes pemrograman, kontes pemrograman lokal ataupun nasional. *Tracking* dapat dimanfaatkan untuk mendapatkan pelajar berprestasi yang dapat dikelompokkan untuk kontes seperti ACM ICPC.

Berdasarkan uraian diatas, maka dikembangkanlah sebuah sistem *Online Judge* yang mampu menyelesaikan hal tersebut. Oleh karena itu, topik tugas akhir ini diangkat dengan judul **“Aplikasi Online Judge dan Tracking Pelajar berbasis Kompetisi Pemrograman”**.

## **1.2 Rumusan Masalah**

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. *Online Judge* yang menyediakan media penambahan bank soal belum tersedia.
2. Sistem untuk *tracking* pelajar belum tersedia pada *Online Judge* yang ada untuk diimplementasi.

## **1.3 Tujuan dan Manfaat**

Tujuan dari pembuatan tugas akhir ini adalah mengembangkan sistem penjurian *online* yang memudahkan pengguna dalam penambahan bank soal sekaligus dapat digunakan untuk *tracking* pelajar.

Manfaat dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Memudahkan penjurian secara *online* dalam perlombaan kompetisi pemrograman seperti ACM, IOI maupun Ideafuse.
2. Sebagai layanan yang bisa digunakan oleh kampus-kampus maupun organisasi-organisasi yang ingin melaksanakan kompetisi pemrograman.
3. Meningkatkan kemampuan peserta tidak hanya pada penyelesaian soal, tetapi mampu menghasilkan soal yang baik.

## **1.4 Batasan Masalah**

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki batasan, yaitu:

1. Bahasa pemrograman yang dilayani oleh sistem adalah C dan C++.
2. Soal yang digunakan bertipe *classical* (standard I/O).

## **1.5 Metodologi Pengembangan Perangkat Lunak**

Untuk memecahkan masalah di atas, maka tugas akhir ini menggunakan metodologi yang sistematis dengan model *waterfall*. Tahapan-tahapannya sebagai berikut:

### **1. *Requirement***

Mengidentifikasi tentang apa yang akan dikerjakan dengan menggunakan metode studi literatur yaitu mencari referensi di perpustakaan maupun di internet.

Tahapan yang akan dilakukan dalam proses *requirement* adalah sebagai berikut :

- Pengumpulan data yang diperlukan dalam perancangan *Online Judge*
- Pengumpulan data *rules* atau aturan yang akan diimplementasikan dalam setiap penggunaan *Online Judge* pada kompetisi pemrograman.

### **2. *Analysis***

Menganalisis perancangan *Online Judge*. Dalam perancangan perangkat lunak tersebut, akan digunakan *tools* pemodelan data UML untuk mempermudah perancangan desain antar muka.

Adapun diagram UML yang akan digunakan adalah sebagai berikut:

- Pemanfaatan *use case* diagram sebagai diagram yang menggambarkan interaksi pengguna dengan perangkat lunak.
- Pemanfaatan *activity* diagram sebagai diagram yang menggambarkan aktifitas dari perangkat lunak.

### **3. *Design***

Melakukan perancangan desain antar muka dan basis data berdasarkan diagram-diagram UML yang telah dianalisis sebelumnya ke dalam bentuk *mockup* yang menjadi dasar perancangan tampilan pada *website* dan basis data yang menjadi struktur penyimpanan data.

#### 4. *Implementation*

Implementasi sistem *Online Judge* dilakukan dalam 2 tahap, sebagai berikut:

a. Pemrograman (*Coding*)

Mengimplementasikan desain yang telah dibuat ke dalam sistem *website*.

b. *Testing*

Pengujian dilakukan sebanyak 2 tahapan:

i. Pengujian Sistem

Melakukan pengujian atau *testing* terhadap sistem *website* yang telah dibuat dan memperbaiki kesalahan selama tahap pengujian. Pengujian dilakukan dengan melihat bahasa yang digunakan telah sesuai, menguji waktu eksekusi (*Time Limit Exceeded*) tidak dilewati dan punya batasan, dan menguji memori yang digunakan (*Memory Limit*).

ii. *User Acceptance Testing*

Pengujian tidak dilakukan sebatas *developer* tetapi dilakukan oleh Mikroskil Programming Club yang merupakan sebuah organisasi di Mikroskil yang bertujuan untuk membina mahasiswa mampu mengikuti kontes pemrograman. Dalam pengujian ini, pengguna mengikuti 2 kontes yang diadakan serta menggunakan fitur lain dari aplikasi.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pemrograman**

Pemrograman adalah proses yang dilakukan terhadap sebuah permasalahan komputasi komputer untuk menghasilkan sebuah program komputer (Kerninghan, 1999). Pemrograman meliputi proses analisis, memahami, menulis, menguji, memperbaiki (*debug*), implementasi serta meningkatkan ketepatan dan mengurangi sumber daya yang digunakan. Tujuan utama dari pemrograman adalah menghasilkan sebuah program yang dapat secara otomatis melaksanakan suatu aktivitas atau menyelesaikan suatu permasalahan. Sebuah program ditulis dalam kode yang harus dimengerti oleh mesin komputer. Kode ini dapat ditulis dalam berbagai bahasa pemrograman.

Bahasa pemrograman memiliki gaya pemrograman yang berbeda satu sama lain. Pemilihan bahasa pemrograman biasanya memiliki banyak pertimbangan, seperti kebijakan bahasa pemrograman, kesesuaian terhadap masalah, ketersediaan serta preferensi tiap individu. Idealnya, bahasa pemrograman ditentukan oleh pengembang aplikasi. Bahasa pemrograman juga dibedakan berdasarkan bahasa tingkat rendah hingga bahasa tingkat tinggi. Bahasa tingkat rendah biasanya lebih susah digunakan tetapi lebih cepat dalam eksekusi program karena *machine-oriented*. Bahasa tingkat tinggi lebih abstrak dan mudah digunakan. Hal ini mengakibatkan banyaknya pengembang perangkat lunak lebih memilih bahasa pemrograman tingkat tinggi daripada bahasa pemrograman tingkat rendah.

Bahasa pemrograman telah menyediakan fungsi pada *library* baik yang disediakan oleh bahasa pemrograman atau pihak ketiga. Fungsi yang disediakan beragam dan dapat digunakan sehingga membantu pengembang aplikasi. Walaupun setiap bahasa memiliki perbedaan, tetapi semua bahasa memiliki intruksi/fungsi dasar yang sama, yaitu:

1. *Input* (masukan): Menerima masukan/data dari pengguna melalui *keyboard*, *file* atau *device*.

2. *Output* (keluaran): Menampilkan data ke layar atau menyimpan data kedalam *file* atau *device*.
3. Aritmatika: Melakukan operasi aritmatika dasar seperti penjumlahan atau pengurangan.
4. Eksekusi bersyarat: Memeriksa sebuah kondisi dan menjalankan pernyataan yang ditentukan dari kondisi tersebut.
5. Perulangan: Melakukan sebuah aksi secara berulang.

Walaupun banyak teknik atau metode dalam pemrograman, program yang baik hendaknya memiliki kriteria mendasar (Hartmann, 2009), seperti:

1. *Efficiency* / kinerja: jumlah sumber daya yang digunakan oleh sebuah program baik waktu proses, memori, *bandwidth* jaringan atau besarnya *space disk* yang digunakan. Semakin sedikit sumber daya yang digunakan maka semakin baik sebuah program.
2. *Reliability*: ketepatan dari sebuah program. Hal ini tergantung pada kebenaran yang diberikan program, kebenaran algoritma, dan kesalahan program yang sedikit seperti kesalahan pada manajemen *resource* (*buffer overflows*) dan kesalahan logika (pembagian terhadap 0).
3. *Robustness*: kemampuan sebuah program mengantisipasi masalah yang terjadi bukan karena kesalahan *programmer*. Masalah yang berupa kesalahan data, data yang *corrupt*, kurangnya sumber daya seperti memori, servis sistem operasi, koneksi jaringan ataupun kesalahan pengguna.
4. *Usability*: bagian ergonomi dari program. Kegunaan dari sebuah program yang menyediakan kemudahan kepada pengguna yang menggunakan untuk tujuan tertentu.
5. *Portability*: Kemampuan sebuah program yang dapat digunakan di perangkat keras komputer yang berbeda atau sistem operasi yang berbeda. Hal ini tergantung kepada fasilitas yang disediakan *platform* baik perangkat keras dan sumber daya sistem operasi.
6. *Maintainability*: kemudahan dari sebuah program yang dapat dimodifikasi oleh pengembang perangkat lunak untuk melakukan perbaikan, penyesuaian, memperbaiki *bug* dan lubang keamanan. Hal ini tidak terlihat

oleh pengguna tetapi dapat secara signifikan mempengaruhi program dalam jangka panjang.

## 2.2 Kompetisi Pemrograman

Kompetisi Pemrograman atau biasa disebut *Competitive Programming* adalah sebuah kompetisi pelatihan logika dan otak yang biasanya diadakan secara luas di Internet atau tempat-tempat tertentu dimana pesertanya mencoba untuk mengirimkan kode program sesuai dengan spesifikasi dari soal yang diberikan (Halim dan Halim, 2013). *Competitive Programming* ini juga banyak didukung dan diadakan oleh beberapa perusahaan ternama seperti Google, Facebook dan IBM. Adapun juga beberapa organisasi ataupun institusi di Indonesia yang juga mengadakan kompetisi sedemikian seperti STMIK Mikroskil, Universitas Indonesia, Universitas Surabaya, Institut Teknologi Bandung, Universitas Bina Nusantara, Universitas Parahyangan, dan sebagainya.

Pada kompetisi pemrograman terdapat beberapa bahasa pemrograman standar yang digunakan seperti C, C++, Java dan Pascal. Bahasa C adalah bahasa yang biasa digunakan oleh sistem operasi Apple. C++ adalah bahasa yang dikembangkan untuk menambahkan fitur berorientasi objek dan penambahan kelas dan fitur dengan bahasa C. Java adalah bahasa pemrograman yang digunakan oleh Google sebagai bagian dari sistem operasi Android. Pascal adalah bahasa yang awalnya digunakan untuk mengajarkan konsep pemrograman komputer tetapi pascal juga digunakan pada Olimpiade Komputer Indonesia (TOKI) dan IOI (*International Olympiad in Informatics*).

Soal yang digunakan pada kompetisi pemrograman bertipe *classical*, yaitu memiliki standar *input* dan *output*. Program yang digunakan untuk menyelesaikan soal pada kompetisi pemrograman akan menerima masukan yang sederhana berupa teks alfabet atau numerik dan memberikan keluaran yang sederhana. Soal bertipe *classical* tidak memiliki GUI (*Graphical User Interface*), melainkan berada pada *command prompt* atau *terminal*.

Setiap soal pada kompetisi pemrograman memiliki komponen sebagai berikut:

1. Deskripsi masalah; mendeskripsikan masalah dalam suatu bahasa (*default English*) yang perlu diselesaikan.
2. Spesifikasi *input-output*; detil format dari masukan dan keluaran yang perlu ditangani oleh program peserta. Keluaran dari program harus sama persis dengan spesifikasi format keluaran. Kesalahan pada presentasi *output* (misalnya terdapat lebih spasi) akan menghasilkan *Wrong Answer*.
3. Sampel *input-output*; beberapa sampel masukan dan keluaran.
4. Batas waktu; waktu maksimal yang diperbolehkan untuk program dijalankan dan menghasilkan keluaran yang diharapkan.

### 2.3 Online Judge

Sistem Penjurian *Online* atau disebut *Online Judge* adalah sebuah sistem *online* untuk menguji suatu program dalam kontes pemrograman (Skeina dan Revilla, 2003). Sistem ini juga biasanya digunakan untuk berlatih pada kontes-kontes yang terkait dengan pemrograman.

Sistem ini dapat mengkompilasi, mengeksekusi kode, dan mengujinya dengan data yang sudah diberikan sebelumnya. Kode yang dikirimkan dijalankan dengan berbagai pembatasan atau pengaturan, termasuk batas waktu, batas memori, pembatasan keamanan dan sebagainya. *Output* dari kode akan ditangkap oleh sistem, dan dibandingkan dengan *output* standar. Sistem kemudian akan mengembalikan hasilnya. Ketika kesalahan ditemukan dalam *output* standar, maka dilakukanlah *rejudgement* atau pengiriman ulang kode dengan menggunakan metode yang sama.

Sistem *Online Judge* juga menyediakan *scoreboard* atau papan urutan peringkat yang menunjukkan urutan pengguna dengan jumlah terbesar dari solusi yang benar dan waktu eksekusi terpendek untuk masalah tertentu. Bagian berikut akan menjelaskan sistem dan implementasi dari *Online Judge* yang ada.

#### 2.3.1 UVa Online Judge

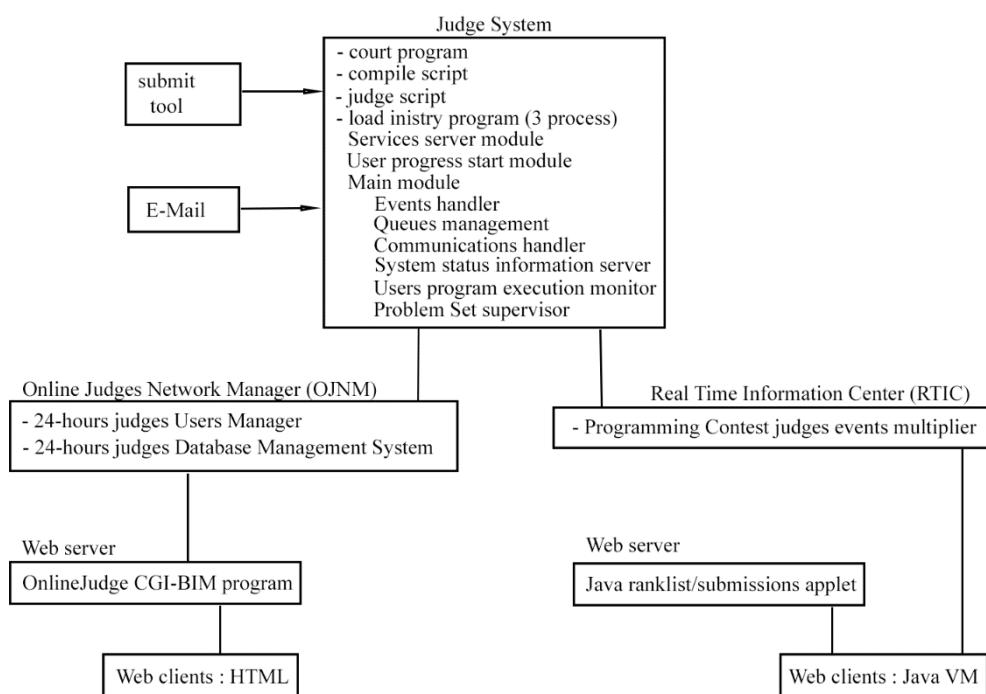
UVa *Online Judge* adalah situs pelatihan kontes pemrograman yang cukup populer dan banyak digunakan (Revilla dkk, 2008). UVa menyediakan sistem

penjurian otomatis dimana pengguna dapat mengirim solusi dari soal yang ada untuk mengecek kebenaran dari solusi sekaligus melatih kemampuan pemrograman.

Sistem pertama yang dibangun dengan standar Unix SH dibangun ulang untuk menambah perbaikan terhadap pengujian sistem. Walaupun begitu, standar Unix tidak cukup kuat untuk mendukung sistem penjurian yang dapat diandalkan, contohnya Unix tidak mampu membatasi memori yang digunakan dari kode program yang akan dijalankan. Oleh karena itu, aplikasi penjurian yang baru dikembangkan kembali dan disesuaikan dengan model ICPC baik dari tipe soal dan aturan yang ada.

Arsitektur aplikasi UVa dirancang untuk mampu menyediakan alat pembelajaran kepada pengguna sekaligus latihan pada kontes. Diagram servis dan aplikasi yang menjadi dasar perencanaan pengembangan UVa Online Judge dapat dilihat pada Gambar 2.1.

## NETJUDGE 2.0 ARCHITECTURE



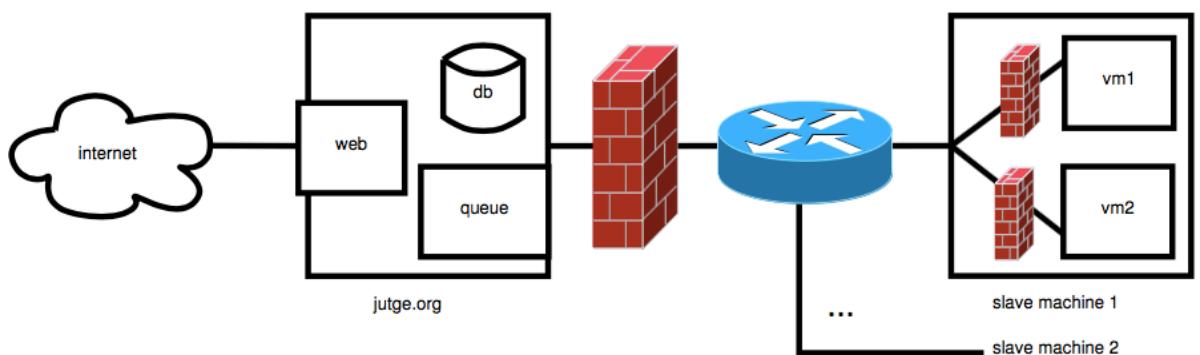
Gambar 2.1 Rancangan sistem UVa Online Judge (Revilla dkk, 2008)

UVa Online Judge mulai dibuka kepada umum dengan ratusan soal dan sedikit promosi pada tanggal 15 April 1997. UVa semakin mulai dikenal publik setelah kontes pemrograman semakin populer. Selama 2 tahun, penjurian UVa tidak mengalami perubahan kecuali perbaikan masalah kecil dan tambahan soal pemrograman yang diambil dari situs lainnya. UVa mengalami perubahan kembali dengan menambah beberapa servis yang digunakan untuk menangani penambahan pengguna dan *submissions* pada penjurian *online*.

### 2.3.2 Jutge.org

Jutge.org adalah penjurian pemrograman *online* yang dapat diakses publik untuk edukasi (Caterineum dan Petit, 2011). Pengguna dapat menyelesaikan soal pemrograman lebih dari 800 soal dengan 22 bahasa pemrograman. Hasil dari solusi ditentukan oleh pengecekan baik waktu, memori hingga keamanan. Dibandingkan dengan penjurian *online* lainnya, Jutge.org dibuat untuk murid dan pengajar. Jutge.org menyediakan lingkungan dimana pengajar dapat membuat pengajaran dan soal untuk murid.

Infrastruktur Jutge.org mengandung 6 *server* Linux. Salah satu *server* menyediakan *front-end* berupa *web*, basis data dan antrian *submissions*. *Server* yang lain merupakan *back-end* dan digunakan sebagai mesin untuk memproses *submissions*. Infrastruktur Jutge.org dapat dilihat pada Gambar 2.2.



Gambar 2.2 Infrastruktur Jutge.org (Caterineum dan Petit, 2011)

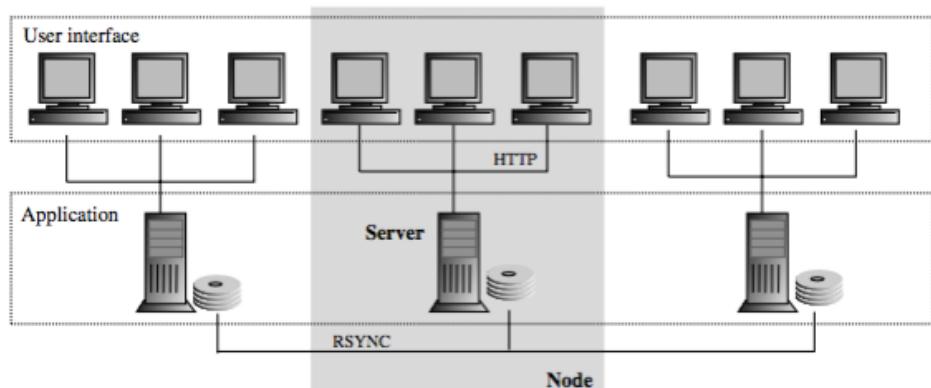
*Server* utama yang memiliki *front-end* langsung terkoneksi dengan internet. Untuk masalah keamanan, *server* lainnya hanya terkoneksi kepada *server* utama

melalui koneksi pribadi sehingga tidak memiliki koneksi internet. *Server* yang melakukan proses penjurian juga memiliki mesin *virtual* yang akan *restart* setiap proses dengan *firewall*.

Untuk menjaga integritas dari sistem, semua eksekusi dari program dilakukan pada lingkungan *sandbox* yang membatasi akses data, pemantauan sistem, dan pembatasan waktu CPU serta memori. Tetapi setiap *restart* akan memakan banyak waktu sehingga proses *restart* akan dilakukan bila telah melakukan penjurian yang cukup banyak, dimana setiap proses akan di-*kill* setelah selesai.

### 2.3.3 Mooshak

Mooshak adalah sistem manajemen kontes pemrograman pada *web*. Mooshak mendukung berbagai jenis kontes, terutama yang sesuai dengan aturan ICPC. Mooshak tidak memiliki *problem set* yang dapat diselesaikan diluar kontes, dikarenakan Mooshak ditujukan untuk kontes. Tetapi penggunaannya semakin luas terutama untuk *programming courses* dan menvalidasi *assignments* perkuliahan. Mooshak diimplementasikan pada sistem operasi Linux dengan *server* HTTP Apache dan bahasa Tcl.



Gambar 2.3 Infrastruktur Mooshak (Leal dan Silva, 2003)

Arsitektur dari Mooshak sama seperti aplikasi *web* kebanyakan, yaitu *client-server framework* yang mengkoneksikan pengguna dengan mesin dimana soal dan *submissions* disimpan, diproses dan divalidasi (Leal dan Silva, 2003). Pada Gambar 2.3 dapat terlihat arsitektur Mooshak yang terbagi atas *layer* horisontal dan vertikal.

Lapisan *user interface* meliputi mesin yang digunakan oleh tim, juri, admin dan pengguna yang mengakses sistem. GUI (*Graphical User Interface*) ditampilkan dengan HTML dan interaksi data dilakukan ke *server* pada *application layer* dengan protokol HTTP. Setiap *application layer* berupa beberapa *server* yang masing-masing menggunakan sistem manajemen data sendiri.

Mooshak juga memiliki *network* sendiri dimana setiap lapisan dari mesin *client* yang terkoneksi dengan *server* menggunakan *nodes*. Suatu kontes pemrograman dapat dibuat dengan sebuah *node* Mooshak. *Server* Mooshak merupakan *server* HTTP Apache yang memiliki program eksternal dengan protokol CGI. *Server* Mooshak dijalankan pada sistem operasi Linux. Apache mengatur komunikasi, otentikasi, akses kontrol dan enkripsi. Program eksternal mengatur interaksi antar muka HTML dan proses data.

## 2.4 Standar Penjurian

Standar penjurian pada *online judge* disesuaikan dengan standar penjurian yang ada pada ACM ICPC. Batasan waktu (*Time Limit*) merupakan waktu eksekusi maksimum yang diizinkan pada sebuah penjurian untuk menghasilkan *output* dari semua *input* yang diuji. Sebuah soal akan dianggap *solved* apabila *submission* yang dikirimkan *accepted* oleh sistem/juri. Untuk menghasilkan *submission* yang *accepted*, sistem akan meng-*compile* dan mengetes program dengan masukan yang hanya dimiliki oleh juri dimana masukan tersebut disediakan sebelum kontes diadakan (Manzoor, 2006).

Setiap *submission* akan mendapatkan salah satu respon dari sistem:

1. *Accepted*

Program yang menghasilkan *output* sesuai dan benar dalam batas waktu tanpa adanya masalah. Program diterima oleh juri dan masalah dianggap selesai.

2. *Wrong Answer*

Program tidak menghasilkan keluaran yang benar.

3. *Runtime Error*

Program gagal (*crashed*) saat dijalankan dengan masukan yang dimiliki juri.

4. *Time Limit Exceed*

Program menggunakan waktu lebih dari batas waktu yang ditentukan untuk menghasilkan keluaran walaupun sukses dijalankan.

5. *Compile Error*

Program (*source code*) tidak dapat dikompilasi/dijalankan.

## 2.5 Tracking Pelajar

*Tracking* adalah sebuah metode pembagian murid sesuai dengan kemampuan akademis yang pernah dipelajari ke dalam grup atau kelas di dalam sebuah sekolah (Hallinan, 1994). Pada sistem *tracking* sekolah, seluruh murid dicatat sejarah kemampuan dan prestasi yang pernah didapat dan dibagi menjadi grup diatas rata-rata, normal ataupun dibawah rata-rata.

Pada beberapa sekolah, murid dengan murid dengan *tracking* yang bagus menentukan kriteria objektif sekolah, sedangkan murid dengan *tracking* yang buruk menentukan apa yang perlu diperbaiki. Dengan adanya catatan sejarah pelajar maka ilmu dan pendidikan yang diperlukan jauh lebih mudah ditentukan. Pelajar dengan *rating* yang tinggi memerlukan ilmu yang lebih (*advance*) sedangkan pelajar dengan *rating* yang rendah perlu dipantau apa yang menyebabkan pelajar tersebut jauh lebih rendah.

Perkuliahan di beberapa negara menerapkan metode ini untuk menentukan mahasiswa mengikuti kelas akademik yang benar dan sesuai. *Tracking* dapat memberikan keuntungan, keuntungan terbesar yaitu *tracking* memungkinkan pengajar mengetahui kemampuan pelajar dari waktu ke waktu. *Tracking* juga mampu mengidentifikasi pelajar yang berbakat sehingga dapat digabungkan ke dalam suatu grup untuk berlaga pada kompetisi yang ada.

## 2.6 Elo Rating

Sistem Elo *rating* adalah metode untuk menghitung tingkat kemampuan relatif pemain pada permainan kompetitif seperti catur (Silver dan Fischer-Baum, 2015). Elo *rating* dinamai sesuai nama penemunya yaitu Arpad Elo, seorang profesor fisika kelahiran Hungaria.

Sistem Elo awalnya diciptakan sebagai perbaikan sistem *rating* catur tetapi juga digunakan sebagai sistem penilaian untuk kompetisi *multiplayer* di sejumlah *video game*, asosiasi sepak bola, basket, *Major League Baseball*, pemrograman kompetitif, *Scrabble*, dan permainan lainnya.

Perbedaan dalam penilaian antara dua pemain berfungsi sebagai prediktor hasil pertandingan. Dua pemain dengan peringkat yang sama yang bermain melawan satu sama lain diharapkan untuk seri. Seorang pemain dengan rating 100 poin lebih besar dari lawan diharapkan untuk menang dengan persentase 64%; jika perbedaannya adalah 200 poin, maka persentase kemenangan yang diharapkan adalah 76%.

Seorang pemain mempunyai sebuah nilai *rating* yang bertambah atau berkurang tergantung pada hasil pertandingan antar pemain. Setelah setiap pertandingan selesai, pemain yang menang mendapatkan poin dari pemain yang kalah. Perbedaan antara peringkat dari pemain yang menang dan yang kalah menentukan jumlah total poin yang diperoleh atau hilang setelah pertandingan. Dalam serangkaian permainan antara pemain dengan *rating* tinggi dan pemain dengan *rating* rendah dinilai, pemain tinggi dinilai diharapkan untuk mencetak kemenangan. Jika pemain dengan *rating* tinggi menang, maka hanya beberapa poin *rating* akan diambil dari pemain rendah dinilai. Namun, jika pemain dengan *rating* lebih rendah menang, maka pemain tersebut akan mendapat *rating* lebih banyak. Pemain dengan *rating* lebih rendah juga akan mendapatkan poin dari pemain dengan *rating* lebih tinggi.

Terdapat beberapa jenis Elo Rating yang digunakan, sebagai berikut:

1. FIDE Rating

FIDE (*Federation Internationale des Echecs*) merupakan *rating* yang digunakan pada catur. Untuk pemain terbaik di dunia, FIDE *rating* merupakan *rating* yang paling penting. FIDE *rating* selalu berubah setiap bulan. Perhitungan FIDE ditentukan oleh hasil permainan catur. Beberapa organisasi catur menamakannya algoritma 400. Menurut algoritma ini, perhitungan dihitung apabila:

- a. Tambahkan 400 ke *rating* apabila lawan kalah.

- b. Kurangi 400 dari *rating* apabila pemain kalah.
- c. Tambahkan *rating* bila permainan berakhir imbang.
- d. Dibagi terhadap jumlah pemain pada permainan.

Contoh bila pemain A menang 2 kali dan kalah 2 kali dapat dilihat pada persamaan 2.1 dan 2.2.

$$\text{Performance rating} = \frac{(w + 400 + x + 400 + y - 400 + z - 400)}{4} \quad (2.1)$$

$$\text{Performance rating} = \frac{[w + x + y + z + 400(2) - 400(2)]}{4} \quad (2.2)$$

Sehingga dapat disederhanakan menjadi persamaan 2.3.

$$\text{Performance rating} = \frac{\text{Total of opponents' ratings} + 400 * (\text{Wins} - \text{Losses})}{\text{Games}} \quad (2.3)$$

Contoh:

- a. Apabila mengalahkan pemain dengan Elo *rating* 1000 dapat dilihat pada persamaan 2.4.

$$\text{Performance rating} = \frac{1000 + 400 * (1)}{1} = 1400 \quad (2.4)$$

- b. Apabila mengalahkan 2 pemain dengan Elo *rating* 1000 dapat dilihat pada persamaan 2.5.

$$\text{Performance rating} = \frac{2000 + 400 * (2)}{2} = 1400 \quad (2.5)$$

- c. Apabila permainan berakhir seri dapat dilihat pada persamaan 2.6.

$$\text{Performance rating} = \frac{1000 + 400 * (0)}{1} = 1000 \quad (2.6)$$

Perhitungan dipengaruhi oleh faktor *k* yang digunakan untuk membantu perhitungan *performance rating* lebih baik. FIDE *rating* menggunakan faktor *k* dengan aturan:

- a. *k* = 40, apabila pemain tidak pernah memiliki *rating* dan telah memainkan 30 permainan catur serta pemain yang telah berumur 18 tahun apabila memiliki *rating* dibawah 2300.
- b. *k* = 20, apabila pemain selalu memiliki *rating* dibawah 2400.
- c. *k* = 10, apabila pemain pernah memiliki *rating* sebesar 2400 atau lebih dan telah memainkan paling tidak 30 permainan catur. Untuk *rating* pemain dengan *k* = 10, tidak akan berubah terkecuali apabila FIDE *rating* mengalami perubahan.

K faktor ini sangat berguna untuk mengurangi inflasi atau deflasi pada *rating*. Pemain yang memiliki *rating* tinggi akan mendapatkan *rating* jauh lebih sulit (Ilyumzhinov, 2008).

## 2. Live Rating

Live Rating diciptakan untuk menutupi kekurangan FIDE *rating* dimana FIDE *rating* berubah sekali dalam sebulan. Sesuai dengan namanya perubahan *rating* dilakukan setiap *rating* pemain menyelesaikan permainan. Live *rating* dihitung berdasarkan FIDE *rating*.

## 3. United States Chess Federation Rating

USCF menggunakan klasifikasi untuk pemain, sebagai berikut:

- a. 0 - 199: Kelas J
- b. 200 - 399: Kelas I
- c. 400 - 599: Kelas H
- d. 600 - 799: Kelas G
- e. 800 - 999: Kelas F
- f. 1000 - 1199: Kelas E
- g. 1200 - 1399: Kelas D
- h. 1400 - 1599: Kelas C
- i. 1600 - 1799: Kelas B
- j. 1800 - 1999: Kelas A
- k. 2000 - 2199: *Expert*
- l. 2200 - 2399: *Master Nasional*
- m. 2400 dan seterusnya: *Senior Master*

Pada dasarnya, pemula memiliki *rating* sebesar 800, pemain dengan kemampuan rata-rata memiliki *rating* sebesar 1600, dan pemain profesional memiliki *rating* sebesar 2400.

Faktor K pada USCF dapat dihitung dengan pembagian 800 terhadap jumlah permainan dari pemain ( $N_e$ ) ditambah dengan permainan yang dilakukan oleh pemain pada sebuah turnamen ( $m$ ), dapat dilihat pada persamaan 2.7.

$$K = 800 / (N_e + m) \quad (2.7)$$

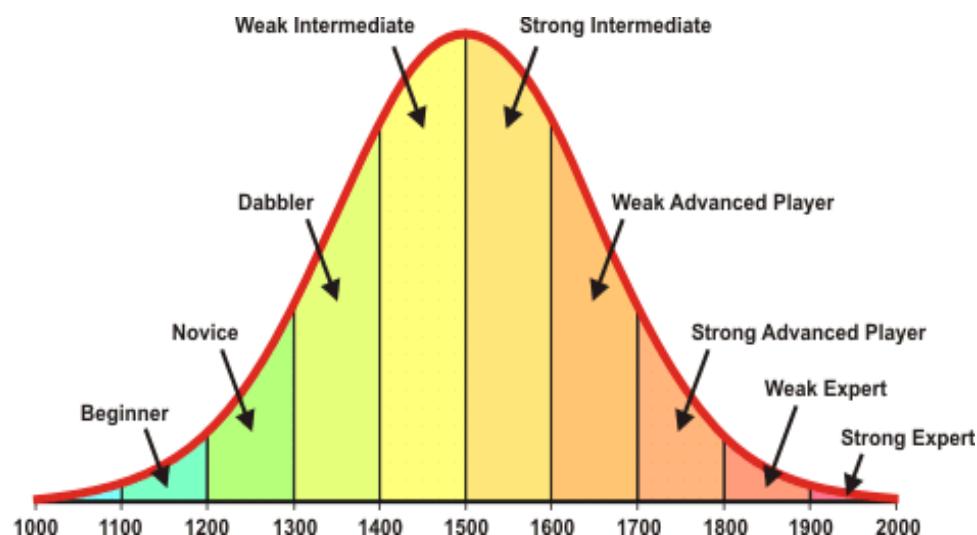
USCF menjamin bahwa *rating* terendah adalah 100, walaupun pemain memiliki performansi yang buruk pada turnamen.

$$AF = \min\{100 + 4N_W + 2N_D + N_R, 150\} \quad (2.8)$$

Pada persamaan 2.8 dapat dilihat *rating* terendah adalah 100 dimana  $N_W$  adalah jumlah permainan yang dimenangkan pemain,  $N_D$  adalah jumlah permainan yang berakhir seri, dan  $N_R$  adalah jumlah turnamen dimana pemain memenangkan tiga atau lebih permainan (Glickman, 2001).

#### 4. Backgammon Pro Rating

*Rating* pada backgammon dimulai dari 1500. Setiap kemenangan akan menaikkan *rating*, begitu juga setiap kekalahan menurunkan *rating*. Grafik kemampuan sebuah pemain terhadap *rating* dapat dilihat dari Gambar 2.4.



Gambar 2.4 Rancangan tampilan input (Silver dan Anthon, 2007)

Faktor yang mempengaruhi perubahan *rating*, yaitu:

- Hasil permainan, kemenangan akan menaikkan *rating*, kekalahan akan menurunkan *rating*. Pemain tidak akan mendapatkan poin *rating* tambahan apabila pemain memenangkan permainan dengan kemenangan yang jauh lebih banyak. Contoh: kemenangan 3-2 terhadap seorang lawan akan memberikan perubahan *rating* yang sama dengan kemenangan 12-0 terhadap seorang lawan.
- Waktu permainan. Semakin lama sebuah permainan akan mempengaruhi *rating* jauh lebih besar, dikarenakan semakin lama

sebuah permainan menghasilkan sebuah permainan yang didasarkan oleh kemampuan pemain bukan didasarkan oleh keberuntungan.

- c. Perbedaan *rating*. Kemenangan terhadap pemain dengan *rating* tinggi akan menaikkan *rating* jauh lebih besar dibandingkan pemain dengan *rating* rendah.
- d. Jumlah permainan yang dilaksanakan.

Persentase kemenangan dihitung dengan persamaan 2.9 dimana D adalah *rating* pemain - *rating* lawan, dan N adalah waktu permainan.

$$P = \frac{1}{1 + 10^{-D\sqrt{N}/2000}} \quad (2.9)$$

Tabel 2.1 Probabilitas kemenangan terhadap *rating*

<b>Rating Diff (D)</b>	<b>Match Length (N)</b>							
	1	3	5	7	9	11	15	25
-500	.360	.270	.216	.179	.151	.129	.097	.053
-400	.387	.311	.263	.228	.201	.178	.144	.091
-300	.415	.355	.316	.286	.262	.241	.208	.151
-200	.443	.402	.374	.352	.334	.318	.291	.240
-100	.471	.450	.436	.424	.415	.406	.390	.360
0	.500	.500	.500	.500	.500	.500	.500	.500
100	.529	.550	.564	.576	.585	.594	.610	.640
200	.557	.598	.626	.648	.666	.682	.709	.760
300	.585	.645	.684	.714	.738	.759	.792	.849
400	.613	.689	.737	.772	.799	.822	.856	.909
500	.640	.730	.784	.821	.849	.871	.903	.947

Berdasarkan probabilitas tersebut, apabila pemain memiliki *rating* dengan 200 poin lebih tinggi dari lawan dan memainkan permainan 9 poin, maka probabilitas kemenangan pemain adalah 66,6%.

Setiap permainan memiliki perbedaan *rating* yang dipengaruhi oleh lama waktu permainan, dihitung dengan persamaan 2.10 dimana N adalah lama permainan.

$$S = 4\sqrt{N} \quad (2.10)$$

Sehingga bila pemainan memenangkan permainan maka poin *rating* yang didapat dihitung dengan persamaan 2.11 dimana P adalah probabilitas kemenangan dan S adalah poin dari permainan.

$$W = (1 - P) * S \quad (2.11)$$

Pada dasarnya semakin besar probabilitas kemenangan maka poin *rating* yang didapat semakin rendah.

Apabila pemain kalah pada permainan, poin *rating* yang dikurangi dihitung dengan persamaan 2.12 dimana P adalah probabilitas kemenangan dan S adalah poin dari permainan (Silver dan Anthon, 2007).

$$L = P * S \quad (2.12)$$

## 5. Elo Rating Codeforces

Elo Rating Codeforces digunakan pada banyak *online judge* dikarenakan sistem *rating* ini terus berkembang dan cocok untuk *rating* peserta (<http://codeforces.com/blog/entry/20762>). Ide dasar dari sistem *rating* Codeforces adalah untuk generalisasi Elo *rating* yang mendukung permainan dengan beberapa peserta. Setiap anggota memiliki sebuah *rating* berupa bilangan bulat, dimana semakin tinggi nilai *rating* mempunyai arti bahwa user memiliki hasil yang lebih baik dalam kontes. *Rating* selalu dikalkulasikan sehingga kesetaraan antar anggota menjadi sesuai yang dapat dilihat pada persamaan 2.13. Pada persamaan 2.13,  $P_{i,j}$  merupakan probabilitas bahwa peserta ke-i memiliki hasil yang lebih baik dibandingkan peserta ke-j. Sehingga probabilitas menang/kalah ditentukan oleh selisih *rating* partisipan.

$$P_{i,j} = \frac{1}{1 + 10^{\frac{r_j - r_i}{400}}} \quad (2.13)$$

Sebagai contoh, apabila perbedaan *rating* setara dengan 200, maka peserta dengan *rating* lebih tinggi seharusnya menang dengan probabilitas 0.75. Apabila perbedaan *rating* setara dengan 400, maka peserta partisipan seharusnya menang dengan probabilitas 0.9. Setelah kontes, nilai *rating* berubah untuk memenuhi rumus Elo.

Lalu posisi yang diharapkan (*seed*) akan dihitung untuk tiap peserta. *Seed* berupa penjumlahan probabilitas kemenangan dari seluruh partisipan dan dihitung dengan persamaan 2.14.

$$seed_i = \sum_{\substack{j=1 \\ j \neq i}}^n P_{j,i} + 1 \quad (2.14)$$

Dengan adanya *seed* maka *rating* akan naik apabila posisi peserta pada kontes lebih baik daripada *seed* dan *rating* akan berkurang apabila posisi peserta lebih buruk daripada *seed*. *Seed* dan posisi aktual pada kontes digunakan untuk dihitung rata-rata geometris. Layaknya rata-rata dari *seed* dan posisi aktual pada kontes mengubah nilai *rating* ke posisi yang benar. Dengan menggunakan *binary search* untuk mencari nilai *rating R* yang sesuai dengan persamaan 2.15.

$$d_i = (R - r_i) / 2 \quad (2.15)$$

Walaupun sistem *rating* sudah cukup bagus, masih terdapat kekurangan yaitu inflasi *rating*. Inflasi membuat peserta dengan *rating* tinggi akan terus naik. Untuk menghindari hal tersebut maka untuk peserta yang memiliki *rating* tertinggi akan digabungkan dalam suatu grup dan dihitung nilai heuristik dengan persamaan 2.16.

$$s = \min(n, 4\sqrt{n}) \quad (2.16)$$

Lalu penjumlahan dari semua nilai  $d_i$  peserta disesuaikan hingga mendapat nilai 0.  $r_i = r_i + inc$ , dimana  $inc = - \frac{\text{sum}_s}{s}$ ,  $\text{sum}$  merupakan penjumlahan dari semua  $d_i$  peserta dari grup tersebut.

Pencapaian *rating* yang konsisten diperlukan menyelesaikan 2 permasalahan, yaitu:

- a. Bila peserta A memiliki *rating* yang lebih rendah dari peserta B sebelum kontes dan apabila selesai kontes peserta A memiliki posisi lebih rendah dari peserta B maka saat perhitungan terjadi, perubahan *rating* peserta A tidak boleh lebih besar dari perubahan *rating* peserta B.
- b. Bila peserta A menyelesaikan kontes lebih baik dari peserta B tetapi A memiliki *rating* lebih buruk dari peserta B sebelum kontes maka peserta

A harus memiliki perubahan *rating* sama atau lebih besar dibandingkan peserta B.

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis**

Proses analisis ini mencakup analisis proses mengenai perhitungan Elo *rating*. Kemudian akan dilanjutkan dengan analisis kebutuhan yang berupa analisis kebutuhan fungsional dengan menggunakan *use case* diagram dan analisis kebutuhan non fungsional.

##### **3.1.1 Analisis Kebutuhan**

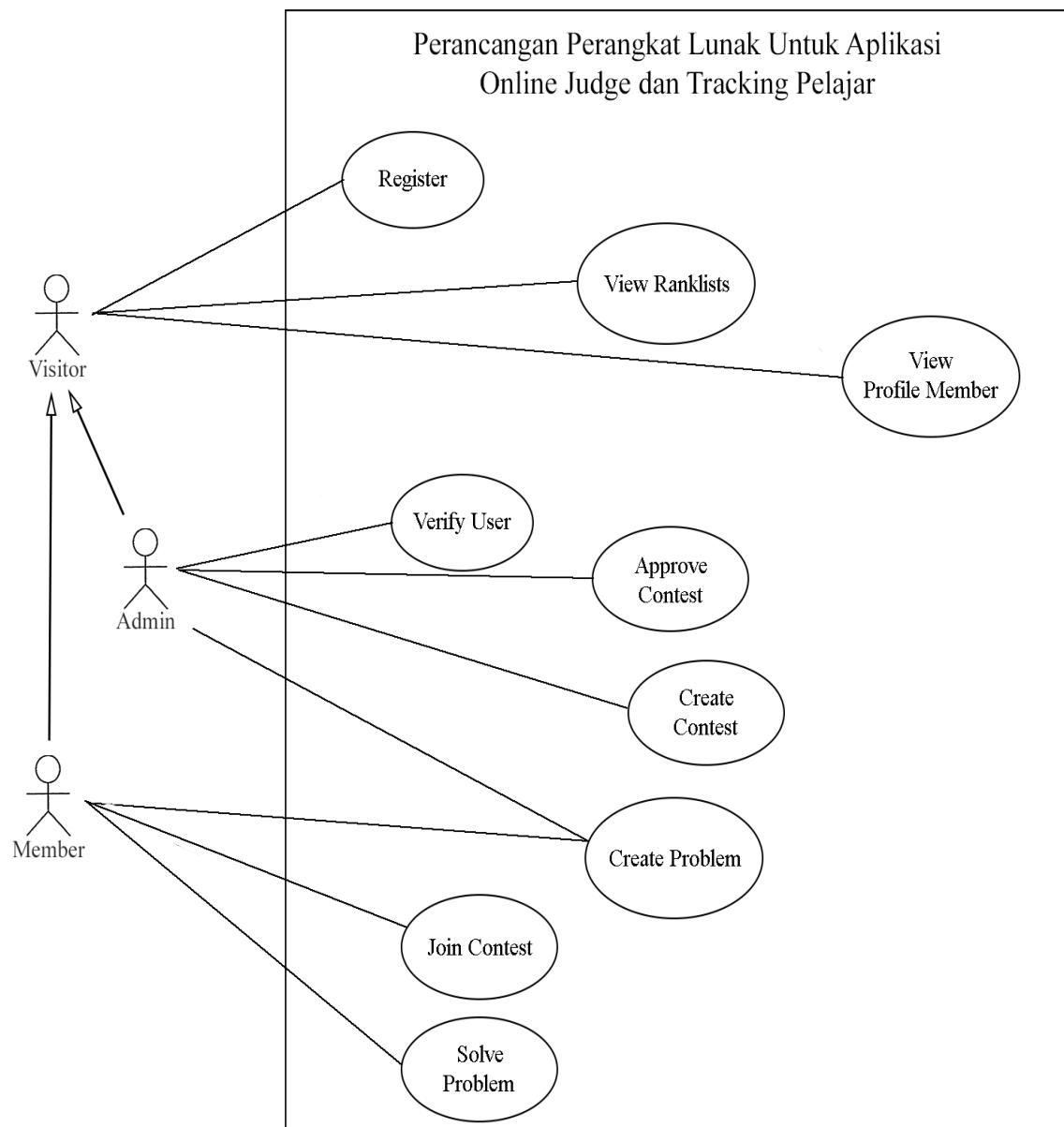
Analisis kebutuhan terhadap sistem yang akan dirancang mencakup analisis kebutuhan fungsional dan analisis kebutuhan non fungsional. Analisis kebutuhan fungsional mendeskripsikan fungsionalitas yang harus dipenuhi oleh perangkat lunak, sedangkan analisis kebutuhan non fungsional mendeskripsikan kualitas dari sistem yang dirancang.

###### **3.1.1.1 Kebutuhan Fungsional**

Adapun beberapa kebutuhan fungsional yang harus dipenuhi oleh perangkat lunak adalah sebagai berikut:

1. Perangkat Lunak menyediakan *user administrator* yang mampu mengatur validitas *member*.
2. Perangkat Lunak mampu menampilkan *history* dan pergerakan *rating member*.
3. Perangkat Lunak dapat memberikan *user administrator* ataupun *member* membuat soal/masalah yang dapat diikutkan pada kontes ataupun *public*.
4. Perangkat Lunak dapat memberikan *user administrator* membuat kontes pemrograman yang diikuti oleh *member*.
5. Perangkat Lunak mampu menghitung pergerakan *rating member* setelah kontes selesai.
6. Perangkat Lunak mampu menampilkan *ranklists member* berdasarkan *rating*.

Analisis kebutuhan fungsional dari perangkat lunak ini dapat digambarkan dengan menggunakan *use case diagram* yang menggambarkan interaksi pengguna dengan sistem. Gambaran interaksi pengguna dengan sistem dalam *use case* diagram dapat dilihat pada Gambar 3.1.



Gambar 3.1 Interaksi Pengguna dengan Sistem dalam *Use Case Diagram*

Narasi dari deskripsi aktor pada Gambar 3.1 dapat dilihat pada Tabel 3.1

Tabel 3.1 Deskripsi Aktor

Nama aktor	Deskripsi
<i>Visitor</i>	orang yang pertama kali menggunakan sistem <i>online judge</i> dan orang yang hanya mengunjungi <i>website</i>
<i>Admin</i>	orang yang bertanggung jawab mengelola sistem <i>online judge</i>
<i>Member</i>	orang yang telah terdaftar dalam sistem <i>online judge</i>

Narasi *Use Case Diagram* pada Gambar 3.1 adalah:

1. Use Case Register

Narasi dari Use Case Register bisa dilihat di Tabel 3.2

Tabel 3.2 Tabel Use Case Register

Nama use case	Register	
Aktor	<i>Visitor</i>	
Deskripsi	Use Case <i>visitor register</i> / daftar pada sistem	
Sasaran	<i>Visitor</i> dapat daftar dan menggunakan sistem	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. <i>Visitor</i> membuka halaman <i>Register</i> lalu mengisikan <i>data user</i> seperti nama, <i>password</i> , <i>email</i> , universitas dan jenis kelamin	2. Sistem akan menyimpan data lalu memberikan notifikasi kepada <i>visitor</i> untuk menunggu <i>admin</i> menvalidasi.
Kesimpulan	Use case <i>visitor</i> daftar pada sistem	
Postkondisi	<i>Data visitor</i> tercatat dan menunggu <i>admin</i> menverifikasi	

2. Use Case View Ranklists

Narasi dari Use Case View Ranklists bisa dilihat di Tabel 3.3

Tabel 3.3 Tabel Use Case View Ranklists

Nama use case	View Ranklists
Aktor	<i>Visitor</i>
Deskripsi	Use case melihat <i>ranklists</i> dari <i>member</i> dan universitas berdasarkan <i>rating</i>
Prakondisi	Terdapat <i>member</i> untuk dapat dilihat <i>ranklists</i>

Sasaran	Menampilkan <i>ranklists member</i> atau universitas berdasarkan <i>rating</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. <i>Visitor</i> membuka halaman <i>ranklists</i> 3. <i>Visitor</i> memilih <i>tab university</i>	2. Sistem menampilkan <i>ranklists member</i> dengan Elo <i>rating</i> tertinggi hingga terendah 4. Sistem menampilkan <i>ranklists universitas</i> dengan <i>rating</i> tertinggi hingga terendah
Kesimpulan	Use case melihat <i>ranklists member</i> dan universitas	
Postkondisi	Tampilan <i>ranklists</i> ditampilkan ke pengguna	

### 3. Use Case View Profile Member

Narasi dari Use Case View Profile Member bisa dilihat di Tabel 3.4

Tabel 3.4 Tabel Use Case View Profile Member

Nama use case	View Profile Member	
Aktor	<i>Visitor</i>	
Deskripsi	Use case menampilkan <i>profile</i> yang dimiliki <i>member</i>	
Prakondisi	<i>Member</i> telah terdaftar dan diverifikasi oleh admin	
Sasaran	<i>Visitor</i> dapat melihat <i>history</i> dan <i>rating member</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. <i>Visitor</i> membuka halaman <i>profile member</i> dari halaman depan web	2. Sistem menampilkan <i>profile member</i> dengan <i>rating member</i> dan <i>history rating member</i>
Kesimpulan	Use case melihat <i>profile member</i>	
Postkondisi	<i>Profile member</i> ditampilkan	

### 4. Use Case Verify User

Narasi dari Use Case Verify User bisa dilihat di Tabel 3.5

Tabel 3.5 Tabel Use Case Verify User

Nama use case	Verify User	
Aktor	Admin	
Deskripsi	Use case menverifikasi <i>user</i> yang telah register	
Prakondisi	Terdapat <i>user</i> yang telah register	
Sasaran	Menverifikasi <i>user</i> agar dianggap <i>valid</i> oleh sistem	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. Admin membuka halaman <i>dashboard</i> . 2. Admin memilih <i>user</i> yang ingin divalidasi dan mengklik tombol <i>Verify</i>	3. Sistem akan melakukan <i>update</i> pada <i>user</i> yang ingin dipilih. 4. Sistem akan menampilkan halaman <i>dashboard</i> yang telah melakukan <i>update</i> pada <i>user</i> tersebut.
Kesimpulan	Use case menverifikasi <i>user</i> yang telah <i>register</i> .	
Postkondisi	<i>User</i> yang telah diverifikasi dapat <i>login</i> dan menggunakan fitur sistem	

## 5. Use Case Approve Contest

Narasi dari Use Case Approve Contest bisa dilihat di Tabel 3.6

Tabel 3.6 Tabel Use Case Approve Contest

Nama use case	Approve Contest	
Aktor	Admin	
Deskripsi	Use case dimana admin meng- <i>approve</i> kontes apabila tidak ada kecurangan pada kontes dan <i>rating</i> akan berubah	
Prakondisi	Kontes telah berjalan dan selesai	
Sasaran	Admin dapat meng- <i>approve</i> kontes dan <i>rating</i> dapat berubah	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. Admin <i>login</i> kedalam sistem 2. Admin membuka halaman Contest List	5. Sistem mengubah Elo <i>rating</i> tiap peserta kontes berdasarkan <i>ranking</i> dari kontes

	3. Admin membuka <i>scoreboard</i> kontes yang ingin di- <i>approve</i> 4. Admin menekan tombol Approve Contest	
Kesimpulan	Use case admin dapat <i>approve</i> kontes dan menghitung <i>rating</i> sesuai <i>ranking</i> dari kontes	
Postkondisi	<i>Rating</i> tiap peserta kontes berubah dan kontes di- <i>approve</i>	

## 6. Use Case Create Contest

Narasi dari Use Case Create Contest bisa dilihat di Tabel 3.7

Tabel 3.7 Tabel Use Case Create Contest

Nama use case	Create Contest	
Aktor	Admin	
Deskripsi	Use case membuat kontes untuk diikuti <i>member</i>	
Prakondisi	Soal telah ada untuk dimasukkan pada <i>problem sets</i> kontes	
Sasaran	Admin dapat membuat kontes	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. Admin <i>login</i> ke dalam sistem 2. Admin masuk ke halaman <i>create contest</i> 4. Admin mengisi data mengenai kontes 6. Admin menambah <i>problem</i> yang perlu diselesaikan pada kontes	3. Sistem menampilkan data yang perlu diisi mengenai kontes 5. Sistem menyimpan data dan menampilkan list kontes yang ada
Kesimpulan	Use case membuat kontes	
Postkondisi	Kontes dapat diisi soal dan dapat diikuti <i>member</i>	

## 7. Use Case Create Problem

Narasi dari Use Case Create Problem bisa dilihat di Tabel 3.8

Tabel 3.8 Tabel Use Case Create Problem

Nama use case	Create Problem	
Aktor	<i>Admin dan Member</i>	
Deskripsi	Use case membuat soal yang dapat diselesaikan secara <i>global</i> atau di kontes	
Prakondisi	<i>Member</i> atau Admin harus login terlebih	
Sasaran	Membuat soal yang digunakan pada kontes atau secara <i>global</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. Admin atau <i>member</i> membuka halaman <i>create problem sets</i> 3. Admin atau <i>member</i> mengisi data tentang soal yang ingin dibuat	2. Sistem menampilkan <i>field problem set</i> yang perlu diisi tentang soal yang akan dibuat 4. Sistem menyimpan data yang telah diisi 5. Soal belum terpublish dan dapat digunakan pada kontes atau dipublish admin
Bidang alternative		alt 5. Soal telah terpublish
Kesimpulan	Use case membuat soal yang digunakan pada kontes atau secara <i>global</i>	
Postkondisi	Soal yang dibuat oleh admin dapat dipublish oleh admin atau digunakan pada kontes, sedangkan soal yang dibuat oleh <i>member</i> langsung dipublish	

#### 8. Use Case Join Contest

Narasi dari Use Case Join Contest bisa dilihat di Tabel 3.9

Tabel 3.9 Tabel Use Case Join Contest

Nama use case	Join Contest	
Aktor	<i>Member</i>	
Deskripsi	Use case <i>member join</i> kontes yang ingin diikuti	
Prakondisi	<i>Member login</i> kedalam sistem	
	Aksi Aktor	Respons Sistem
Bidang khas suatu sistem	1. <i>Member</i> membuka halaman kontes	3. Sistem menampilkan kontes telah diikuti

	2. <i>Member</i> menekan tombol Join Contest pada kontes yang ingin diikuti	dan akan dibuka saat waktu telah dimulai
Kesimpulan	Use case mengikuti kontes	
Postkondisi	Kontes diikuti <i>member</i> dan kontes tersebut akan mempengaruhi <i>rating member</i> sesuai dengan performansi <i>member</i>	

## 9. Use Case Solve Problem

Narasi dari Use Case Solve Problem bisa dilihat di Tabel 3.10

Tabel 3.10 Tabel Use Case Solve Problem

Nama use case	Solve Problem	
Aktor	<i>Member</i>	
Deskripsi	Use case <i>member</i> menyelesaikan masalah / <i>problem</i>	
Prakondisi	Terdapat soal yang ter-published	
	Aksi Aktor	Respons Sistem
Bidang khas suatu sistem	1. <i>Member</i> login ke dalam sistem 3. <i>Member</i> membuka halaman <i>Problems</i> untuk melihat daftar soal 5. <i>Member</i> memilih soal yang ingin diselesaikan 7. <i>Member</i> mengirim kode program berbentuk file ke dalam sistem	2. Sistem menampilkan halaman depan website 4. Sistem menampilkan daftar soal yang ada 6. Sistem menampilkan deskripsi mengenai soal yang akan diselesaikan 8. Sistem memberikan status dan hasil mengenai kode program
Bidang alternative	alt 7. Member mengetik kode program pada kotak yang ada dan mengirim kode setelah selesai	
Kesimpulan	Use case menyelesaikan masalah / <i>problem</i>	

Postkondisi	Terdapat data yang tercatat apakah <i>member</i> telah menyelesaikan masalah / <i>problem</i> dengan benar atau salah
-------------	---

### 3.1.1.2 Kebutuhan Non Fungsional

Kebutuhan non-fungsional dari sistem adalah sebagai berikut :

1. Desain Perangkat Lunak dapat ditampilkan secara konsisten pada semua jenis *browser*.
2. Perangkat Lunak dapat berfungsi dengan cepat dan akurat apabila terdapat banyak kompilasi program sewaktu kontes berlangsung.

### 3.1.2 Analisis Proses

Analisis Proses terhadap sistem yang akan dirancang mencakup perhitungan Elo Rating yang dijelaskan dengan contoh pengkalkulasian *rating*. Analisis tentang proses cara kerja algoritma Elo Rating dalam mengkalkulasikan *rating* peserta kontes ketika kontes sudah berakhir. Berikut ini merupakan contoh pengkalkulasian *rating* dengan ilustrasi 5 peserta masing-masing diberikan nilai *rating* 1500 sesuai dengan tabel 3.11 menggunakan Elo Rating.

Tabel 3.11 Tabel Scoreboard

Rank	Name	Solved	Score (Menit)	Rating Awal
1	Andi	4	120	1599
2	Budi	3	80	1571
3	Cindi	3	90	1690
3	Doni	3	90	1659
4	Edi	2	90	1634

Langkah-langkah algoritmanya sebagai berikut:

1. Kalkulasikan *seed* untuk setiap kontestan

*Seed* adalah sebuah nilai probabilitas ranking yang seharusnya didapat oleh masing-masing peserta di dalam kontes tersebut berdasarkan nilai *rating* dirinya sendiri yang dibandingkan dengan kontestan lainnya. Untuk menghitung nilai *seed*, *rating* 1 peserta dibandingkan dengan semua peserta terkecuali *rating* dirinya sendiri dengan menggunakan Elo Rating. Untuk menghitung *seed* tersebut digunakanlah persamaan 3.1 dan persamaan 3.2

dimana  $R_i$  adalah *rating* kontestan yang mau dibandingkan dan  $R_j$  adalah *rating* kontestan yang dibandingkan dengan kontestan  $R_i$ .

$$P_{i,j} = \frac{1}{1 + 10^{\left(\frac{(R_j - R_i)}{400}\right)}} \quad (3.1)$$

$$seed_i = \sum_{j=1, j \neq i}^n P_{i,j} + 1 \quad (3.2)$$

Perhitungan *seed* untuk kontestan Andi yaitu:

$$P_{1,2} = \frac{1}{1 + 10^{\left(\frac{(1571 - 1599)}{400}\right)}} \quad P_{1,3} = \frac{1}{1 + 10^{\left(\frac{(1690 - 1599)}{400}\right)}} \quad P_{1,4} = \frac{1}{1 + 10^{\left(\frac{(1659 - 1599)}{400}\right)}}$$

$$P_{1,2} = \frac{1}{1 + 10^{\left(\frac{-28}{400}\right)}} \quad P_{1,3} = \frac{1}{1 + 10^{\left(\frac{91}{400}\right)}} \quad P_{1,4} = \frac{1}{1 + 10^{\left(\frac{60}{400}\right)}}$$

$$P_{1,2} = 0.5402 \quad P_{1,3} = \frac{1}{1 + 10^{\left(\frac{91}{400}\right)}} \quad P_{1,4} = \frac{1}{1 + 10^{\left(\frac{60}{400}\right)}}$$

$$P_{1,3} = 0.3719 \quad P_{1,4} = 0.4145$$

$$P_{1,5} = \frac{1}{1 + 10^{\left(\frac{(1634 - 1599)}{400}\right)}}$$

$$P_{1,5} = \frac{1}{1 + 10^{\left(\frac{35}{400}\right)}}$$

$$P_{1,5} = 0.4498$$

Perhitungan setiap Probabilitas kontestan dapat dilihat pada tabel 3.12.

Tabel 3.12 Tabel Probabilitas kontestan

Rj Ri	Andi	Budi	Cindi	Doni	Edi
Andi	X	0.5402	0.3719	0.4145	0.4498
Budi	0.4598	X	0.3351	0.3760	0.4103
Cindi	0.6280	0.6649	X	0.5445	0.5799
Doni	0.5855	0.6240	0.4555	X	0.5359
Edi	0.5502	0.5897	0.4201	0.4641	X

$$seed_1 = (P_{1,2} + P_{1,3} + P_{1,4} + P_{1,5}) + 1$$

$$seed_1 = (0.5402 + 0.3719 + 0.4145 + 0.4498) + 1$$

$$seed_1 = 2.7764$$

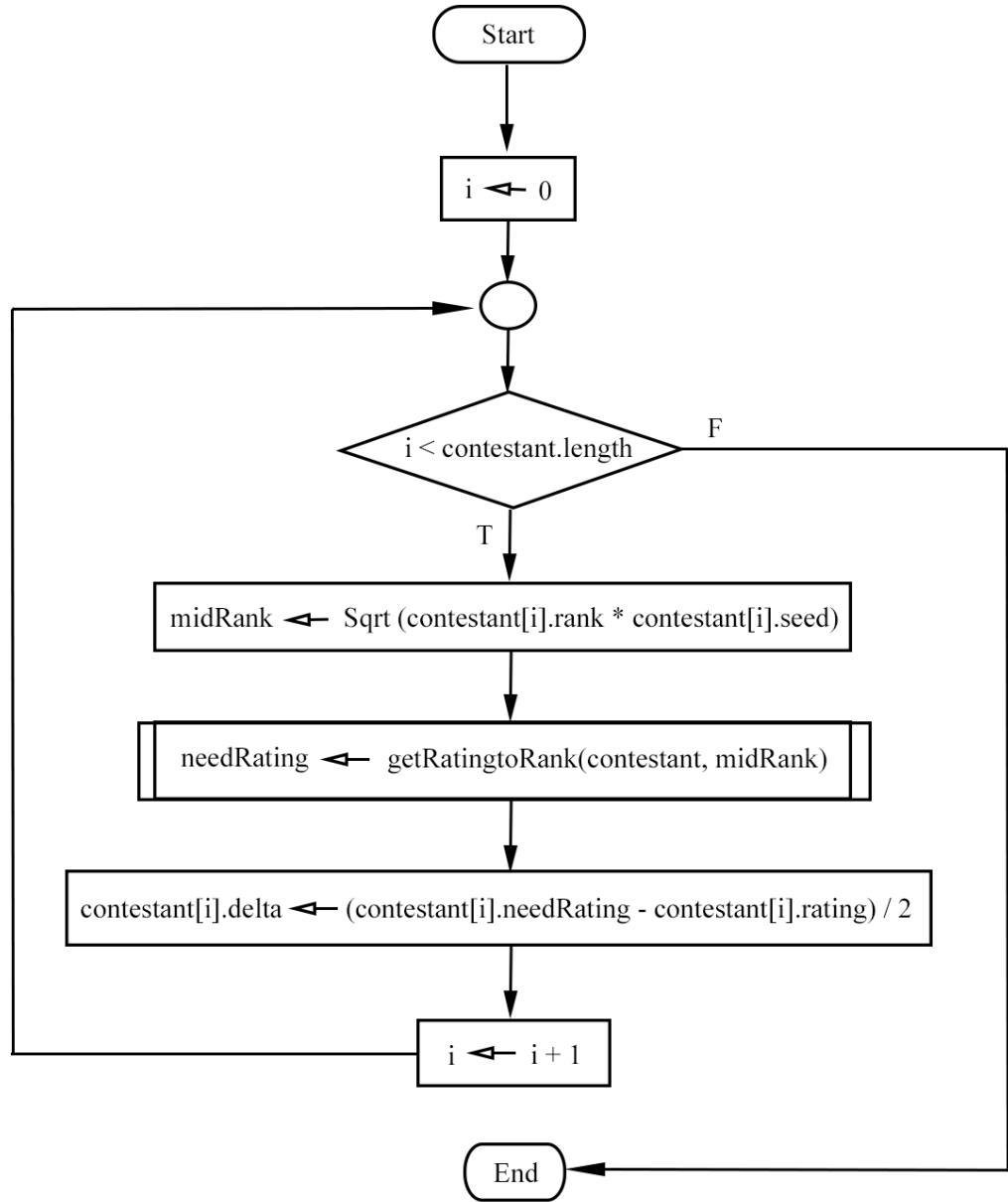
Hasil dari *seed* tiap peserta dapat dilihat pada Tabel 3.13.

Tabel 3.13 Tabel Seed

Rank	Name	Solved	Score (Menit)	Rating Awal	Seed
1	Andi	4	120	1599	2.7764
2	Budi	3	80	1571	2.5812
3	Cindi	3	90	1690	3.4173
3	Doni	3	90	1659	3.2010
4	Edi	2	90	1634	3.0241

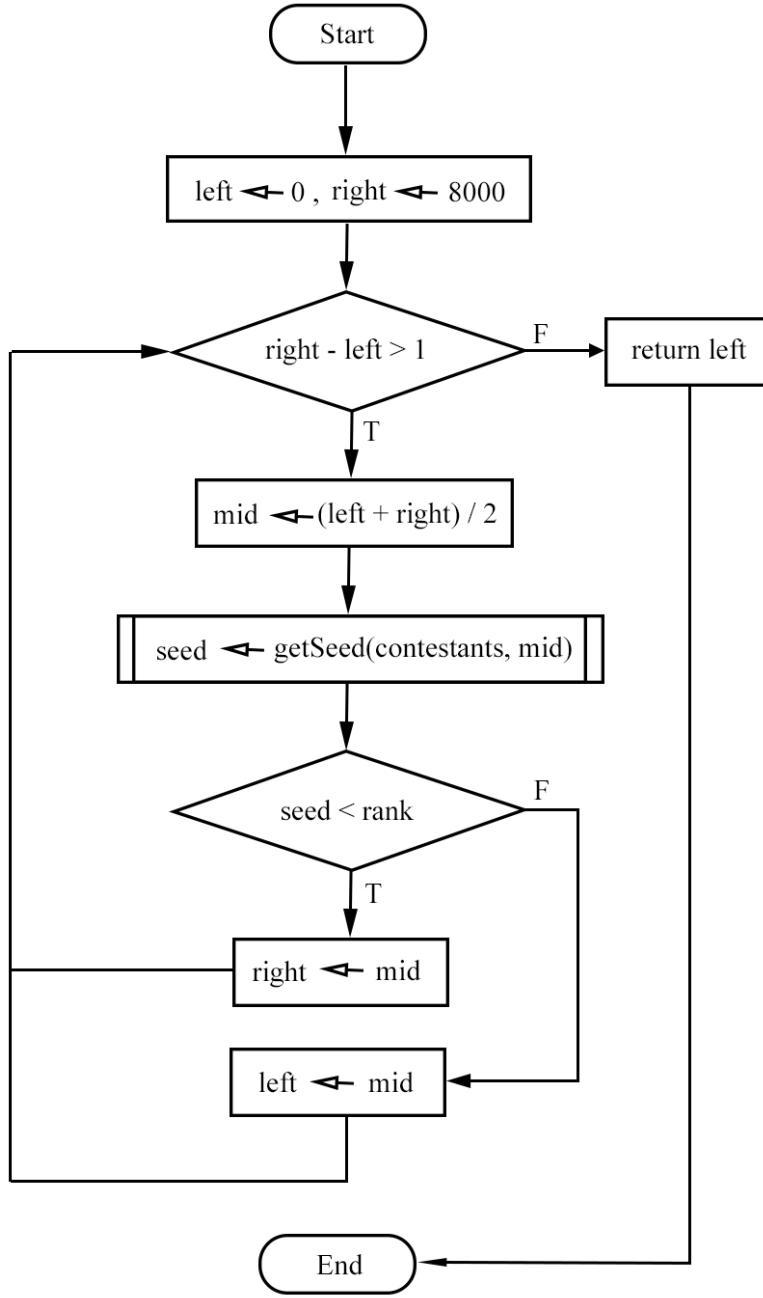
## 2. Kalkulasikan delta untuk setiap kontestan

Delta adalah sebuah nilai acuan bagi bertambahnya atau berkurangnya nilai *rating* pada setiap kontestan. Untuk menghitung delta maka dikalkulasikan terlebih dahulu nilai *rating* yang seharusnya didapatkan oleh masing-masing kontestan berdasarkan *Ranking*, *Solved*, *Score* dan *Rating Awal* masing-masing. Untuk mendapatkan nilai *rating* yang seharusnya didapatkan, digunakanlah sebuah algoritma *Binary Search* dengan fungsi yang dapat dilihat pada Gambar 3.2.



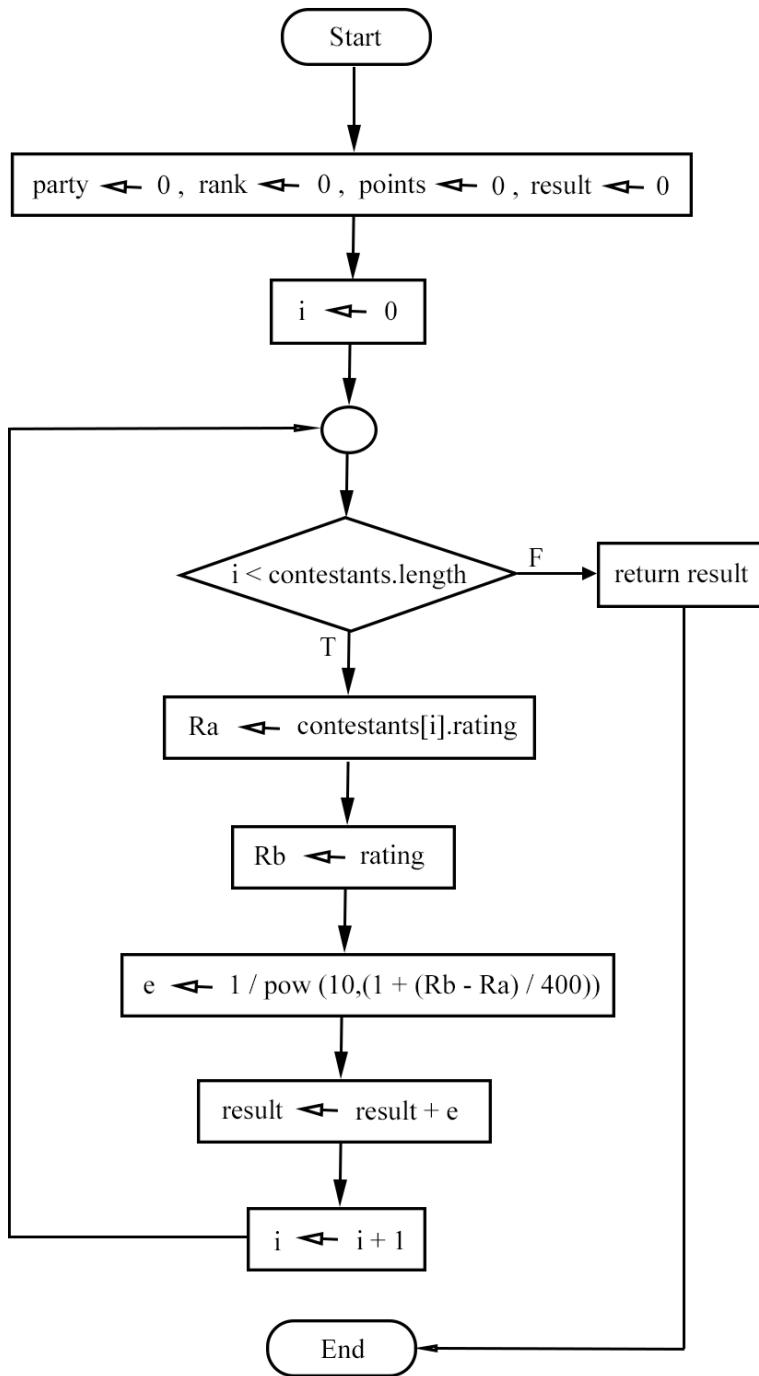
Gambar 3.2 Flowchart *Binary Search rating*

Pada flowchart gambar 3.2, terdapat sub-fungsi berupa `getRatingtoRank` yang menerima 2 parameter yaitu `contestant` dan `midRank`. Fungsi dari `getRatingtoRank` dapat dilihat pada gambar 3.3.



Gambar 3.3 Flowchart `getRatingtoRank`

Pada flowchart gambar 3.3, terdapat sub-fungsi berupa `getSeed` yang menerima 2 parameter yaitu `contestants` dan `mid`. Fungsi dari `getSeed` dapat dilihat pada gambar 3.4.



Gambar 3.4 Flowchart getSeed

Flowchart diatas bertujuan untuk melakukan kalkulasi terlebih dahulu midRank dari masing-masing setiap kontestan persamaan 3.3.

$$midRank_i = \sqrt{Rank_i * Seed_i} \quad (3.3)$$

Kemudian, midRank tersebut dibandingkan dengan *seed* kontestan ekstra yang diberikan nilai konstan 0 menggunakan algoritma Binary Search untuk mendapatkan nilai *rating* yang seharusnya didapatkan oleh kontestan. Kemudian dikalkulasikan delta dengan menggunakan persamaan 3.4.

$$d_i = \frac{\text{needRating}_i - \text{Rating}_i}{2} \quad (3.4)$$

Perhitungan delta untuk kontestan Andi :

$$\text{midRank}_1 = \sqrt{1 * 2.7764}$$

$$\text{midRank}_1 \approx 1.67$$

Kemudian setelah mendapatkan midRank, dilakukanlah perhitungan berikut untuk mendapatkan nilai needRating :

- Lakukan perulangan dengan nilai acuan jika nilai right – left < 1 maka perulangan berhenti. (Left = 0; Right = 8000)
- Carilah terlebih dahulu mid *rating* dari nilai left dan right tersebut.

$$\text{Mid} = (\text{left} + \text{right}) / 2$$

$$\text{Mid} = (0 + 8000) / 2$$

$$\text{Mid} = 4000.5$$

- Kemudian dengan mid tersebut, dapatkan nilai *seed* dengan acuan dari nilai mid tersebut dan jumlahkan nilai *seed* yang didapatkan tersebut.

$$E_{1,dummy} = \frac{1}{1+10^{\frac{(1599-4005.5)}{400}}} \quad E_{1,dummy} \approx 0.000000991$$

$$E_{2,dummy} = \frac{1}{1+10^{\frac{(1571-4005.5)}{400}}} \quad E_{2,dummy} \approx 0.000000844$$

$$E_{3,dummy} = \frac{1}{1+10^{\frac{(1690-4005.5)}{400}}} \quad E_{3,dummy} \approx 0.000001674$$

$$E_{4,dummy} = \frac{1}{1+10^{\frac{(1659-4005.5)}{400}}} \quad E_{4,dummy} \approx 0.000001400$$

$$E_{5,dummy} = \frac{1}{1+10^{\frac{(1634-4005.5)}{400}}} \quad E_{5,dummy} \approx 0.000001213$$

$$\begin{aligned} \text{Sum} &= 0.000000991 + 0.000000844 + 0.000001674 + \\ &0.000001400 + 0.000001213 \end{aligned}$$

$$\text{Sum} \approx 0.00000612$$

- d. Bandingkan nilai total *seed* tersebut dengan midRank awal = 1.67. Apabila nilai total *seed* lebih kecil daripada midRank tersebut maka nilai right = mid, sebaliknya jika nilai total *seed* lebih besar atau sama dengan midRank maka nilai left = mid.

$$\text{Sum} = 0.00000612$$

$$\text{Rank} = 1.67$$

$$\text{Sum} < \text{Rank}$$

$$\therefore \text{Right} = 4000.5$$

- e. Lakukan kembali perhitungan nomor 1, hingga memenuhi aturan right – left > 1

Perulangan ke – 2 :  $(4000.5 - 0 > 1)$

$$\text{Mid} = (0 + 4000.5) / 2$$

$$\text{Mid} = 2000.75$$

$$E_{1,dummy} = \frac{1}{1+10^{\left(\frac{(1599-2000.75)}{400}\right)}} \quad E_{1,dummy} \approx 0.090079968$$

$$E_{2,dummy} = \frac{1}{1+10^{\left(\frac{(1571-2000.75)}{400}\right)}} \quad E_{2,dummy} \approx 0.0777125729$$

$$E_{3,dummy} = \frac{1}{1+10^{\left(\frac{(1690-2000.75)}{400}\right)}} \quad E_{3,dummy} \approx 0.1432173584$$

$$E_{4,dummy} = \frac{1}{1+10^{\left(\frac{(1659-2000.75)}{400}\right)}} \quad E_{4,dummy} \approx 0.1226822955$$

$$E_{5,dummy} = \frac{1}{1+10^{\left(\frac{(1634-2000.75)}{400}\right)}} \quad E_{5,dummy} \approx 0.1080146630$$

$$\text{Sum} = 0.090079968 + 0.0777125729 + 0.1432173584 +$$

$$0.1226822955 + 0.1080146630$$

$$\text{Sum} \approx 0.54$$

$$\text{Sum} = 0.54$$

$$\text{Rank} = 1.67$$

$$\text{Sum} < \text{Rank}$$

$$\therefore \text{Right} = 2000.75$$

Perulangan ke – 3 :  $(2000.75 - 0 > 1)$

$$\text{Mid} = (0 + 2000.75) / 2$$

$$\text{Mid} = 1000.875$$

$$E_{1,dummy} = \frac{1}{1+10^{(\frac{(1599-1000.875)}{400})}} \quad E_{1,dummy} \approx 0.96902422$$

$$E_{2,dummy} = \frac{1}{1+10^{(\frac{(1571-1000.875)}{400})}} \quad E_{2,dummy} \approx 0.96380274$$

$$E_{3,dummy} = \frac{1}{1+10^{(\frac{(1690-1000.875)}{400})}} \quad E_{3,dummy} \approx 0.98142013$$

$$E_{4,dummy} = \frac{1}{1+10^{(\frac{(1659-1000.875)}{400})}} \quad E_{4,dummy} \approx 0.97787063$$

$$E_{5,dummy} = \frac{1}{1+10^{(\frac{(1634-1000.875)}{400})}} \quad E_{5,dummy} \approx 0.97453264$$

$$\text{Sum} = 0.96902422 + 0.96380274 + 0.98142013 + 0.97787063 + 0.97453264$$

$$\text{Sum} \approx 4.86$$

$$\text{Sum} = 4.86$$

$$\text{Rank} = 1.67$$

$$\text{Sum} > \text{Rank}$$

$$\therefore \text{Left} = 1000.875$$

Perulangan ke – 4 :  $(2000.75 - 1000.875 > 1)$

$$\text{Mid} = (1000.875 + 2000.75) / 2$$

$$\text{Mid} = 1500.8125$$

$$E_{1,dummy} = \frac{1}{1+10^{(\frac{(1599-1500.8125)}{400})}} \quad E_{1,dummy} \approx 0.63$$

$$E_{2,dummy} = \frac{1}{1+10^{(\frac{(1571-1500.8125)}{400})}} \quad E_{2,dummy} \approx 0.59$$

$$E_{3,dummy} = \frac{1}{1+10^{(\frac{(1690-1500.8125)}{400})}} \quad E_{3,dummy} \approx 0.74$$

$$E_{4,dummy} = \frac{1}{1+10^{(\frac{(1659-1500.8125)}{400})}} \quad E_{4,dummy} \approx 0.71$$

$$E_{5,dummy} = \frac{1}{1+10^{(\frac{(1634-1500.8125)}{400})}} \quad E_{5,dummy} \approx 0.68$$

$$\text{Sum} = 0.63 + 0.59 + 0.74 + 0.71 + 0.68$$

*Sum*  $\approx$  3.38

Sum = 3.38

Rank = 1.67

Sum > Rank

$\therefore$  Left = 1500.8125

Perulangan ke - 5 :  $(2000.75 - 1500.8125 > 1)$

Mid =  $(1500.8125 + 2000.75) / 2$

Mid = 1750.78125

$$E_{1,dummy} = \frac{1}{1+10^{\left(\frac{(1599-1750.78125)}{400}\right)}} \quad E_{1,dummy} \approx 0.29$$

$$E_{2,dummy} = \frac{1}{1+10^{\left(\frac{(1571-1750.78125)}{400}\right)}} \quad E_{2,dummy} \approx 0.26$$

$$E_{3,dummy} = \frac{1}{1+10^{\left(\frac{(1690-1750.78125)}{400}\right)}} \quad E_{3,dummy} \approx 0.41$$

$$E_{4,dummy} = \frac{1}{1+10^{\left(\frac{(1659-1750.78125)}{400}\right)}} \quad E_{4,dummy} \approx 0.37$$

$$E_{5,dummy} = \frac{1}{1+10^{\left(\frac{(1634-1750.78125)}{400}\right)}} \quad E_{5,dummy} \approx 0.33$$

Sum =  $0.29 + 0.26 + 0.41 + 0.37 + 0.33$

*Sum*  $\approx$  1.67

Sum = 1.67

Rank = 1.67

Sum = Rank

$\therefore$  Left = 1750.78125

Perulangan ke - 6 :  $(2000.75 - 1750.78125 > 1)$

Mid =  $(1750.78125 + 2000.75) / 2$

Mid = 1875.765625

$$E_{1,dummy} = \frac{1}{1+10^{\left(\frac{(1599-1875.765625)}{400}\right)}} \quad E_{1,dummy} \approx 0.16$$

$$E_{2,dummy} = \frac{1}{1+10^{\left(\frac{(1571-1875.765625)}{400}\right)}} \quad E_{2,dummy} \approx 0.14$$

$$E_{3,dummy} = \frac{1}{1+10^{\left(\frac{(1690-1875.765625)}{400}\right)}} \quad E_{3,dummy} \approx 0.25$$

$$E_{4,dummy} = \frac{1}{1+10^{\left(\frac{(1659-1875.765625)}{400}\right)}} \quad E_{4,dummy} \approx 0.22$$

$$E_{5,dummy} = \frac{1}{1+10^{\left(\frac{(1634-1875.765625)}{400}\right)}} \quad E_{5,dummy} \approx 0.19$$

$$Sum = 0.16 + 0.14 + 0.25 + 0.22 + 0.19$$

$$Sum \approx 0.99$$

$$Sum = 0.99$$

$$Rank = 1.67$$

$$Sum < Rank$$

$$\therefore Right = 1875.765625$$

$$\text{Perulangan ke - 7 : } (1875.765625 - 1750.78125 > 1)$$

$$Mid = (1750.78125 + 1875.765625) / 2$$

$$Mid = 1813.2734375$$

$$E_{1,dummy} = \frac{1}{1+10^{\left(\frac{(1599-1813.2734375)}{400}\right)}} \quad E_{1,dummy} \approx 0.22$$

$$E_{2,dummy} = \frac{1}{1+10^{\left(\frac{(1571-1813.2734375)}{400}\right)}} \quad E_{2,dummy} \approx 0.19$$

$$E_{3,dummy} = \frac{1}{1+10^{\left(\frac{(1690-1813.2734375)}{400}\right)}} \quad E_{3,dummy} \approx 0.32$$

$$E_{4,dummy} = \frac{1}{1+10^{\left(\frac{(1659-1813.2734375)}{400}\right)}} \quad E_{4,dummy} \approx 0.29$$

$$E_{5,dummy} = \frac{1}{1+10^{\left(\frac{(1634-1813.2734375)}{400}\right)}} \quad E_{5,dummy} \approx 0.26$$

$$Sum = 0.22 + 0.19 + 0.32 + 0.29 + 0.26$$

$$Sum \approx 1.31$$

$$Sum = 1.31$$

$$Rank = 1.67$$

$$Sum < Rank$$

$$\therefore Right = 1813.2734375$$

$$\text{Perulangan ke - 8 : } (1813.2734375 - 1750.78125 > 1)$$

$$Mid = (1750.78125 + 1813.2734375) / 2$$

$$Mid = 1782.02734375$$

$$E_{1,dummy} = \frac{1}{1+10^{(\frac{(1599-1782.02734375)}{400})}} \quad E_{1,dummy} \approx 0.25$$

$$E_{2,dummy} = \frac{1}{1+10^{(\frac{(1571-1782.02734375)}{400})}} \quad E_{2,dummy} \approx 0.22$$

$$E_{3,dummy} = \frac{1}{1+10^{(\frac{(1690-1782.02734375)}{400})}} \quad E_{3,dummy} \approx 0.37$$

$$E_{4,dummy} = \frac{1}{1+10^{(\frac{(1659-1782.02734375)}{400})}} \quad E_{4,dummy} \approx 0.32$$

$$E_{5,dummy} = \frac{1}{1+10^{(\frac{(1634-1782.02734375)}{400})}} \quad E_{5,dummy} \approx 0.29$$

$$Sum = 0.25 + 0.22 + 0.37 + 0.32 + 0.29$$

$$Sum \approx 1.48$$

$$Sum = 1.48$$

$$Rank = 1.67$$

$$Sum < Rank$$

$$\therefore Right = 1782.02734375$$

Perulangan ke – 9 :  $(1782.02734375 - 1750.78125) > 1$ )

$$Mid = (1750.78125 + 1782.02734375) / 2$$

$$Mid = 1766.404296875$$

$$E_{1,dummy} = \frac{1}{1+10^{(\frac{(1599-1766.404296875)}{400})}} \quad E_{1,dummy} \approx 0.27$$

$$E_{2,dummy} = \frac{1}{1+10^{(\frac{(1571-1766.404296875)}{400})}} \quad E_{2,dummy} \approx 0.24$$

$$E_{3,dummy} = \frac{1}{1+10^{(\frac{(1690-1766.404296875)}{400})}} \quad E_{3,dummy} \approx 0.39$$

$$E_{4,dummy} = \frac{1}{1+10^{(\frac{(1659-1766.404296875)}{400})}} \quad E_{4,dummy} \approx 0.35$$

$$E_{5,dummy} = \frac{1}{1+10^{(\frac{(1634-1766.404296875)}{400})}} \quad E_{5,dummy} \approx 0.31$$

$$Sum = 0.27 + 0.24 + 0.39 + 0.35 + 0.31$$

$$Sum \approx 1.58$$

$$Sum = 1.58$$

$$Rank = 1.67$$

$$Sum < Rank$$

$\therefore \text{Right} = 1766.404296875$

Perulangan ke – 10 :  $(1766.404296875 - 1750.78125) > 1$

$$\text{Mid} = (1750.78125 + 1766.404296875) / 2$$

$$\text{Mid} = 1758.5927734375$$

$$E_{1,dummy} = \frac{1}{1+10^{\frac{(1599-1758.5927734375)}{400}}} E_{1,dummy} \approx 0.28$$

$$E_{2,dummy} = \frac{1}{1+10^{\frac{(1571-1758.5927734375)}{400}}} E_{2,dummy} \approx 0.25$$

$$E_{3,dummy} = \frac{1}{1+10^{\frac{(1690-1758.5927734375)}{400}}} E_{3,dummy} \approx 0.40$$

$$E_{4,dummy} = \frac{1}{1+10^{\frac{(1659-1758.5927734375)}{400}}} E_{4,dummy} \approx 0.36$$

$$E_{5,dummy} = \frac{1}{1+10^{\frac{(1634-1758.5927734375)}{400}}} E_{5,dummy} \approx 0.32$$

$$\text{Sum} = 0.28 + 0.25 + 0.40 + 0.36 + 0.32$$

$$\text{Sum} \approx 1.62$$

$$\text{Sum} = 1.62$$

$$\text{Rank} = 1.67$$

$$\text{Sum} < \text{Rank}$$

$\therefore \text{Right} = 1758.5927734375$

Perulangan ke – 11 :  $(1758.5927734375 - 1750.78125) > 1$

$$\text{Mid} = (1750.78125 + 1758.5927734375) / 2$$

$$\text{Mid} \approx 1754.68$$

$$E_{1,dummy} = \frac{1}{1+10^{\frac{(1599-1754.68)}{400}}} E_{1,dummy} \approx 0.28$$

$$E_{2,dummy} = \frac{1}{1+10^{\frac{(1571-1754.68)}{400}}} E_{2,dummy} \approx 0.25$$

$$E_{3,dummy} = \frac{1}{1+10^{\frac{(1690-1754.68)}{400}}} E_{3,dummy} \approx 0.40$$

$$E_{4,dummy} = \frac{1}{1+10^{\frac{(1659-1754.68)}{400}}} E_{4,dummy} \approx 0.36$$

$$E_{5,dummy} = \frac{1}{1+10^{\frac{(1634-1754.68)}{400}}} E_{5,dummy} \approx 0.33$$

$$Sum = 0.28 + 0.25 + 0.40 + 0.36 + 0.33$$

$$Sum \approx 1.65$$

$$Sum = 1.65$$

$$Rank = 1.67$$

$$Sum < Rank$$

$$\therefore Right = 1754.68$$

$$\text{Perulangan ke - 12 : } (1754.68 - 1750.78125) > 1$$

$$Mid = (1750.78125 + 1754.68) / 2$$

$$Mid \approx 1752.73$$

$$E_{1,dummy} = \frac{1}{1+10^{\left(\frac{(1599-1752.73)}{400}\right)}} \quad E_{1,dummy} \approx 0.29$$

$$E_{2,dummy} = \frac{1}{1+10^{\left(\frac{(1571-1752.73)}{400}\right)}} \quad E_{2,dummy} \approx 0.25$$

$$E_{3,dummy} = \frac{1}{1+10^{\left(\frac{(1690-1752.73)}{400}\right)}} \quad E_{3,dummy} \approx 0.41$$

$$E_{4,dummy} = \frac{1}{1+10^{\left(\frac{(1659-1752.73)}{400}\right)}} \quad E_{4,dummy} \approx 0.36$$

$$E_{5,dummy} = \frac{1}{1+10^{\left(\frac{(1634-1752.73)}{400}\right)}} \quad E_{5,dummy} \approx 0.33$$

$$Sum = 0.29 + 0.25 + 0.41 + 0.36 + 0.33$$

$$Sum \approx 1.67$$

$$Sum = 1.67$$

$$Rank = 1.67$$

$$Sum = Rank$$

$$\therefore Left = 1752.73$$

$$\text{Perulangan ke - 13 : } (1754.68 - 1752.73) > 1$$

$$Mid = (1752.73 + 1754.68) / 2$$

$$Mid \approx 1753.71$$

$$E_{1,dummy} = \frac{1}{1+10^{\left(\frac{(1599-1753.71)}{400}\right)}} \quad E_{1,dummy} \approx 0.29$$

$$E_{2,dummy} = \frac{1}{1+10^{\left(\frac{(1571-1753.71)}{400}\right)}} \quad E_{2,dummy} \approx 0.25$$

$$E_{3,dummy} = \frac{1}{1+10^{\left(\frac{(1690-1753.71)}{400}\right)}} \quad E_{3,dummy} \approx 0.40$$

$$E_{4,dummy} = \frac{1}{1+10^{\left(\frac{(1659-1753.71)}{400}\right)}} \quad E_{4,dummy} \approx 0.36$$

$$E_{5,dummy} = \frac{1}{1+10^{\left(\frac{(1634-1753.71)}{400}\right)}} \quad E_{5,dummy} \approx 0.33$$

$$Sum = 0.29 + 0.25 + 0.40 + 0.36 + 0.33$$

$$Sum \approx 1.66$$

$$Sum = 1.67$$

$$Rank = 1.67$$

$$Sum < Rank$$

$$\therefore Right = 1753.71$$

Perulangan ke – 14 :  $(1753.71 - 1752.73 < 1)$

Perulangan Berhenti

- f. Masukkan nilai left sebagai nilai needRating dari kontestan

$$needRating_1 = left$$

$$needRating_1 \approx 1752.73$$

- g. Hitung delta dengan menggunakan persamaan 3.4

$$d_1 = \frac{1752.73 - 1599}{2}$$

$$d_1 \approx 76.8671$$

Hasil delta dapat dilihat pada tabel 3.14.

Tabel 3.14 Tabel Delta Kontestan

Ran k	Name	Solved	Rating Awal	Seed	NeedRating	Delta
1	Andi	4	1599	2.7764	1752.73	76.8671
2	Budi	3	1571	2.5812	1661.92	45.4626
3	Cindi	3	1690	3.4173	1528.15	-80.9236
3	Doni	3	1659	3.2010	1543.77	-57.6121
4	Edi	2	1634	3.0241	1484.21	-74.8935

3. Jumlahkan semua delta dari setiap kontestan dan hitung Increment rata-rata dari semua delta yang dikalkulasikan

Hasil kalkulasi delta dari setiap peserta kemudian dijumlahkan dan dicarilah increment rata-rata dari kontes tersebut berdasarkan delta dengan menggunakan persamaan 3.5.

$$inc = \frac{-1 * \sum_{i=1}^n d_i}{n-1} \quad (3.5)$$

Dari rata-rata inc tersebut, tambahkan increment tersebut ke setiap delta dari para kontestan dengan persamaan 3.6.

$$d_i = d_i + inc \quad (3.6)$$

Perhitungan penjumlahan delta untuk kontestan Andi :

$$inc = \frac{-1 * (76.8671 + 45.4626 - 80.9236 - 57.6121 - 74.8935)}{5 - 1}$$

$$inc = \frac{-1 * (-91.0994)}{4}$$

$$inc = 22.7748$$

$$d_1 = 76.8671 + 22.7748$$

$$d_1 = 99.6419$$

$$d_1 \approx 99.64$$

Hasil dari Delta dapat dilihat pada tabel 3.15.

Tabel 3.15 Tabel Delta dengan Increment Rata-rata

Rank	Name	Solve d	Rating Awal	Seed	NeedRating	Delta
1	Andi	4	1599	2.7764	1752.73	99.6419
2	Budi	3	1571	2.5812	1661.92	68.2374
3	Cindi	3	1690	3.4173	1528.15	-58.1487
3	Doni	3	1659	3.2010	1543.77	-34.8372
4	Edi	2	1634	3.0241	1484.21	-52.1186

4. Kalkulasi ulang nilai delta sehingga nilai rating tidak mengalami inflasi  
Fungsi dari langkah ini adalah untuk meminimalisir nilai delta yang tidak seimbang misalnya pengurangan yang terlalu banyak atau penambahan

yang terlalu banyak. Kalkulasi diawali dengan menghitung nilai heuristik dari banyak kontestan tersebut dengan menggunakan persamaan 3.7 dimana n adalah jumlah kontestan.

$$h = \min(4 * \sqrt{n}, n) \quad (3.7)$$

Kemudian jumlahkan delta dari setiap kontestan dengan menggunakan perulangan dari nilai h tersebut yang dapat dilihat pada persamaan 3.8.

$$sum = \sum_{i=0}^{h-1} delta_i \quad (3.8)$$

Hasil dari penjumlahan tersebut kemudian dikalkulasikan dengan menggunakan persamaan 3.9 untuk mendapatkan increment.

$$inc = \min \left( \max \left( \frac{-1 * sum}{h}, -10 \right), 0 \right) \quad (3.9)$$

Hasil dari kalkulasi increment tersebut kemudian ditambahkan ke delta masing-masing kontestan sesuai dengan persamaan 3.10.

$$delta_i = delta_i + inc \quad (3.10)$$

Perhitungan delta dengan increment untuk kontestan Andi :

$$h = \min(4 * \sqrt{5}, 5)$$

$$h = \min(4 * 2.23, 5)$$

$$h = \min(8.92, 5)$$

$$h = 5$$

$$sum = -58.1487 - 34.8372 - 52.1186 + 99.6419 + 68.2374$$

$$sum \approx 22.77$$

$$inc = \min \left( \max \left( \frac{-22.77}{5}, -10 \right), 0 \right)$$

$$inc = \min(\max(-4.55, -10), 0)$$

$$inc = \min(-4.55, 0)$$

$$inc = -4.55$$

$$delta_1 = 99.64 - 4.55$$

$$delta_1 = 95.09$$

Hasil dari Delta dapat dilihat pada tabel 3.16.

Tabel 3.16 Tabel Kalkulasi Delta

Rank	Name	Solved	Rating Awal	Seed	NeedRating	Delta
1	Andi	4	1599	2.7764	1752.73	95.09
2	Budi	3	1571	2.5812	1661.92	63.68
3	Cindi	3	1690	3.4173	1528.15	-62.70
3	Doni	3	1659	3.2010	1543.77	-39.39
4	Edi	2	1634	3.0241	1484.21	-56.67

5. Kalkulasikan nilai *rating* dengan delta

Tambahkan nilai *rating* awal terhadap delta yang sudah didapatkan sehingga didapatkan nilai *rating* baru untuk setiap kontestan sesuai dengan persamaan 3.11.

$$newRating_i = Rating\ awal + delta \quad (3.11)$$

Perhitungan *rating* baru untuk kontestan Andi :

$$newRating_1 = 1599 + 95.09 \approx 1694.09$$

Hasil dari *Rating* tiap kontestan dapat dilihat pada tabel 3.17.

Tabel 3.17 Tabel New Rating

Rank	Name	Solved	Rating Awal	Need Rating	Delta	New Rating
1	Andi	4	1599	1752.73	95.09	1694.09
2	Budi	3	1571	1661.92	63.68	1634.68
3	Cindi	3	1690	1528.15	-62.70	1627.30
3	Doni	3	1659	1543.77	-39.39	1619.61
4	Edi	2	1634	1484.21	-56.67	1577.33

### 3.2 Perancangan

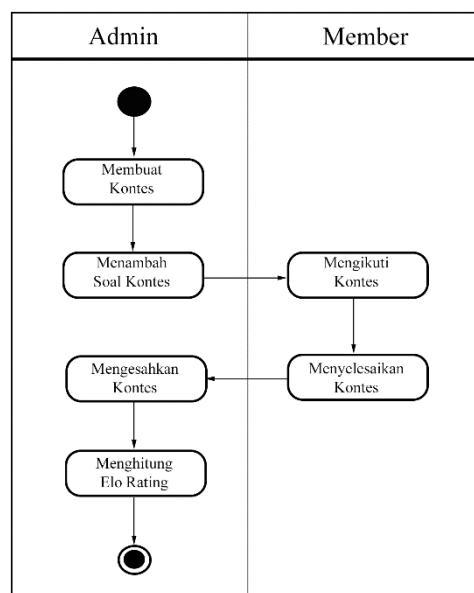
Perancangan pada sistem *online judge* mencakup perancangan sistem usulan dengan *activity diagram*. Kemudian perancangan tampilan di tampilkan dengan *mockup* sedangkan perancangan basis data dengan *class diagram*.

### 3.2.1 Perancangan Sistem Usulan

Perancangan Sistem Usulan dari *online judge* dapat digambarkan dengan *activity diagram*. Activity Diagram dimulai dengan mempartisi bagan menurut aktor-aktor yang terlibat dalam sistem. Kemudian memetakan *workflow*-nya, kapan dimulai, pada bagan aktor yang mana, aktifitas apa saja yang terjadi selama *workflow* dan bagaimana urutan kejadianya sampai dengan berhentinya *workflow*.

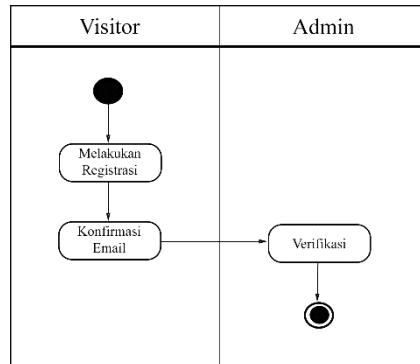
Berikut *activity diagram* dari sistem *online judge*:

#### 1. Activity Diagram Melaksanakan Kontes



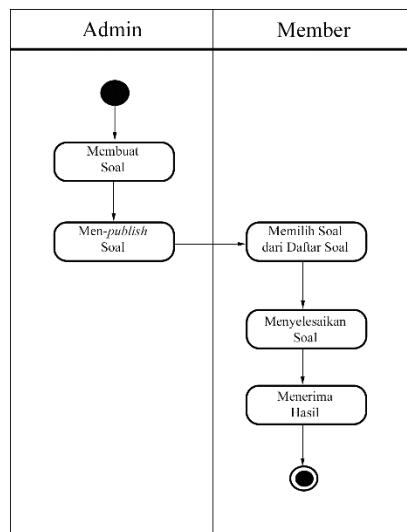
Gambar 3.5 Activity Diagram Melaksanakan Kontes

2. Activity Diagram Verifikasi *User*



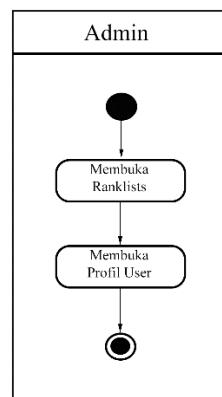
Gambar 3.6 Activity Diagram Verifikasi *User*

3. Activity Diagram Menyelesaikan Soal untuk Admin



Gambar 3.7 Activity Diagram Menyelesaikan Soal

4. Activity Diagram *Tracking* Pelajar



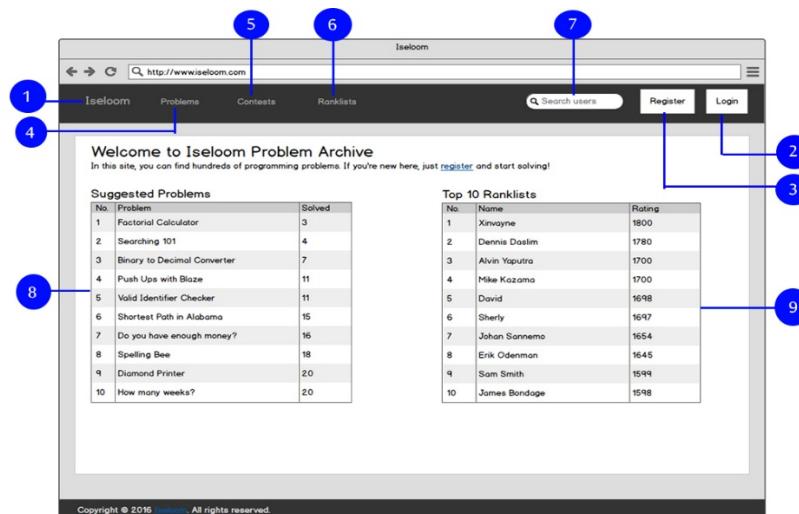
Gambar 3.8 Activity Diagram *Tracking* Pelajar

### 3.2.2 Perancangan Tampilan

Perancangan tampilan / antar muka merupakan tahapan untuk membuat tampilan atau *design* dari sistem yang akan dibuat. Rancangan tampilan yang dibuat meliputi rancangan struktur menu, rancangan *input* dan rancangan *output* dari sistem yang akan dibuat.

#### A. Perancangan Tampilan Halaman Utama

Gambar 3.9 menunjukkan perancangan halaman utama dari sistem. Pengguna dapat melihat beberapa soal pemrograman dan *member* dengan *rating* tertinggi.



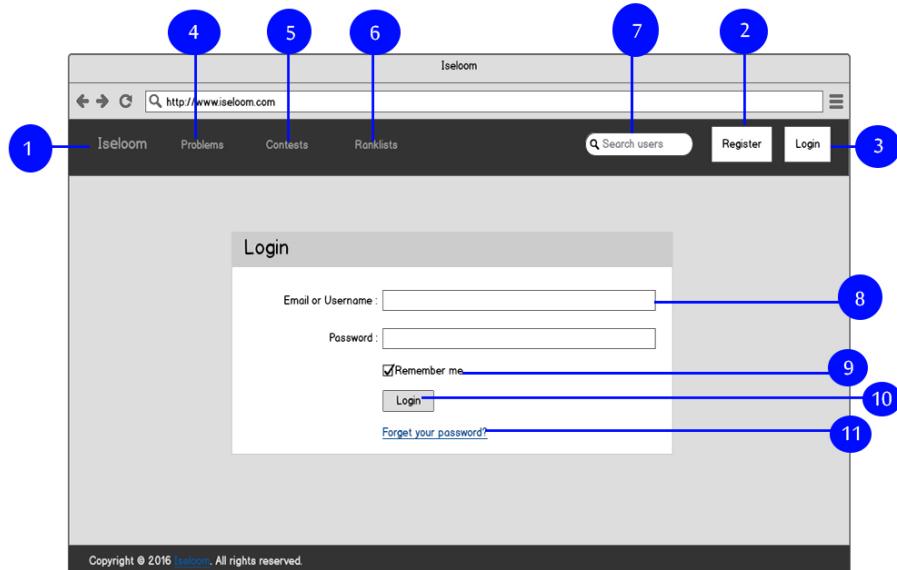
Keterangan:

1. Logo sistem *Online Judge*
2. *Button* “Login” untuk masuk menggunakan akun yang sudah ada
3. *Button* “Register” untuk menuju ke Prosedur Pendaftaran
4. *Link* “Problems” untuk menuju Halaman daftar soal
5. *Link* “Contests” untuk menuju Halaman daftar kontes
6. *Link* “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
8. Daftar 10 soal yang disarankan oleh sistem
9. Daftar 10 pengguna dengan *rating* tertinggi dalam sistem

Gambar 3.9 Perancangan Tampilan Halaman Utama

## B. Perancangan Tampilan Halaman Login

Pada Gambar 3.10, pengguna dapat masuk ke dalam sistem apabila memiliki keanggotaan sebagai *member* atau admin.



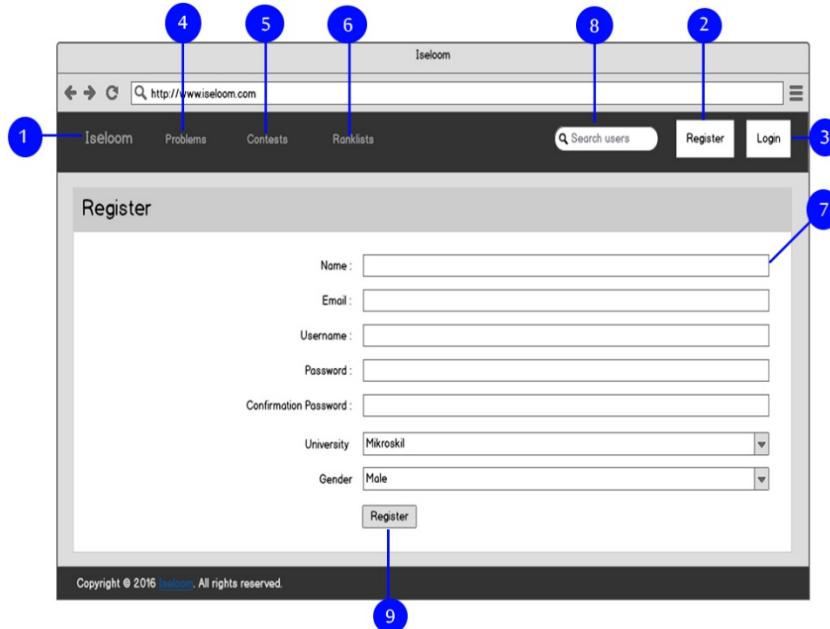
Keterangan:

1. Logo sistem *Online Judge*
2. Button “Register” untuk menuju ke Prosedur Pendaftaran
3. Button “Login” untuk masuk menggunakan akun yang sudah ada
4. Link “Problems” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. SearchBox untuk melakukan pencarian pengguna berdasarkan nama
8. TextBox untuk menuliskan data pengguna untuk *login*
9. CheckBox untuk menyimpan data pengguna untuk jangka waktu panjang
10. Button “Login” untuk melakukan Prosedur *Login*
11. Link “Forget your password?” untuk memasuki Prosedur Lupa Kata Sandi

Gambar 3.10 Perancangan Tampilan Halaman Login

### C. Perancangan Tampilan Halaman Register

Pada Gambar 3.11, pengguna dapat mendaftar diri pada aplikasi untuk dapat menggunakan aplikasi sebagai *member*.



Keterangan:

1. Logo sistem *Online Judge*
2. Button “Register” untuk menuju ke Prosedur Pendaftaran
3. Button “Login” untuk masuk menggunakan akun yang sudah ada
4. Link “Problems” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. TextBox untuk mengetikkan data-data yang diminta
8. SearchBox untuk melakukan pencarian pengguna berdasarkan nama
9. Button “Register” untuk menyimpan data-data yang telah diisi

Gambar 3.11 Perancangan Tampilan Halaman Register

#### D. Perancangan Tampilan Halaman Ranklists

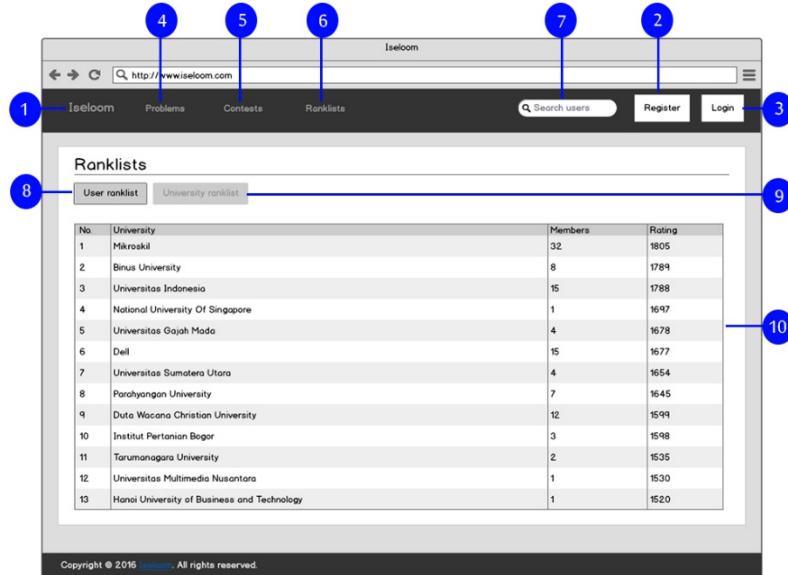
Pada Gambar 3.12 dan Gambar 3.13, pengguna dapat melihat peringkat berdasarkan *rating* tertinggi member ataupun universitas dari tertinggi hingga terendah.

No	User	University	Rating
1	Ximwayne	Mikroskil	1800
2	Dennia Daslim	Mikroskil	1780
3	Alvin Yoputra	Binus University	1700
4	Mike Kazama	Mikroskil	1700
5	David	Mikroskil	1698
6	Sherly	National University Of Singapore	1697
7	Johan Sannemo	Universitas Indonesia	1654
8	Erik Odenman	Binus University	1645
9	Sam Smith	Universitas Indonesia	1599
10	James Bondage	Universitas Indonesia	1598
11	Denny Sutomo	Binus University	1535
12	Anabelle	Universitas Indonesia	1520
13	Dendimon	Universitas Indonesia	1520

Keterangan:

1. Logo sistem *Online Judge*
2. Button “Register” untuk menuju ke Prosedur Pendaftaran
3. Button “Login” untuk masuk menggunakan akun yang sudah ada
4. Link “Problems” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. Button “User ranklist” untuk mengurutkan daftar berdasarkan *rating* pengguna
8. Button “University ranklist” untuk mengurutkan daftar berdasarkan *rating* universitas
9. Tabel daftar *ranklists* berdasarkan pengguna
10. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama

Gambar 3.12 Perancangan Tampilan Halaman Ranklists User



#### Keterangan:

1. Logo sistem *Online Judge*
2. Button “Register” untuk menuju ke Prosedur Pendaftaran
3. Button “Login” untuk masuk menggunakan akun yang sudah ada
4. Link “Problems” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. SearchBox untuk melakukan pencarian pengguna berdasarkan nama
8. Button “User ranklist” untuk mengurutkan daftar berdasarkan *rating* pengguna
9. Button “University ranklist” untuk mengurutkan daftar berdasarkan *rating* universitas
10. Tabel daftar *ranklists* berdasarkan universitas

Gambar 3.13 Perancangan Tampilan Halaman Ranklists Universitas

## E. Perancangan Tampilan Halaman Profil

Pada Gambar 3.14, pengguna dapat melihat profil dari *member* yang ada.

Profil mencakup nama, universitas serta *rating* dari *user* tersebut.



Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Grafik / *History rating* dari pengguna

Gambar 3.14 Perancangan Tampilan Halaman Profil

## F. Perancangan Tampilan Halaman Submissions

Pada Gambar 3.15, member dapat melihat *submission* yang pernah dilakukannya terhadap soal pemrograman dan hasil dari *submission*.

No.	Problem	Date Time	Result
1	123	Thu Apr 28 2016 16:36:27	CORRECT
2	ABC	Thu Apr 28 2016 16:35:27	SUBMISSION ERROR

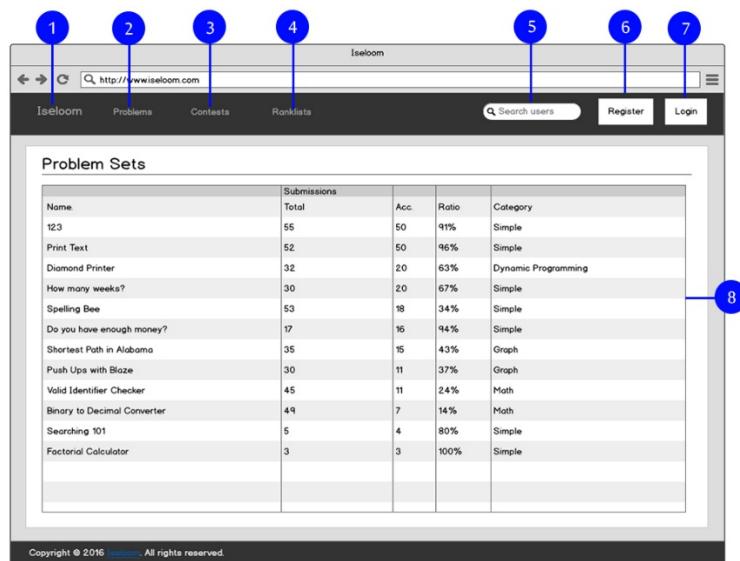
Keterangan:

1. Logo sistem *Online Judge*
2. Link “Mike” untuk menuju Halaman Profil dengan pengguna bernama Mike
3. Link “Submissions” untuk menuju Halaman *submissions* pengguna
4. Link “Problem Sets” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. SearchBox untuk melakukan pencarian pengguna berdasarkan nama
8. Link “Logout” untuk keluar sebagai pengguna dari sistem
9. Tabel daftar *submissions* yang pernah dilakukan pengguna

Gambar 3.15 Perancangan Tampilan Halaman Submissions

## G. Perancangan Tampilan Halaman Problem Sets Member

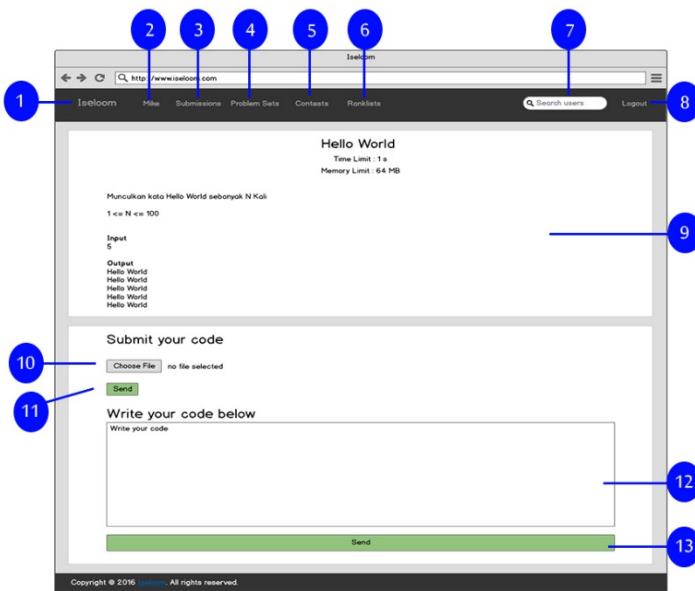
Pada Gambar 3.16, *member* dapat melihat soal yang ada dan dapat diselesaikan tanpa harus mengikuti kontes. Pada Gambar 3.17, *member* mengirim kode program yang digunakan untuk menyelesaikan soal pemrograman berdasarkan deskripsi. *Member* dapat mengirim kode program dalam bentuk *file* ataupun mengetik kode program pada *field* yang telah ada. Pada Gambar 3.18, *member* dapat menghasilkan soal pemrograman dengan mengisikan data mengenai soal pemrograman yang ingin dibuat.



Keterangan:

1. Logo sistem *Online Judge*
2. *Link* “Problems” untuk menuju Halaman daftar soal
3. *Link* “Contests” untuk menuju Halaman daftar kontes
4. *Link* “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
5. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
6. *Button* “Register” untuk menuju ke Prosedur Pendaftaran
7. *Button* “Login” untuk masuk menggunakan akun yang sudah ada
8. Tabel daftar soal dengan kategori serta *submissions* terhadap soal

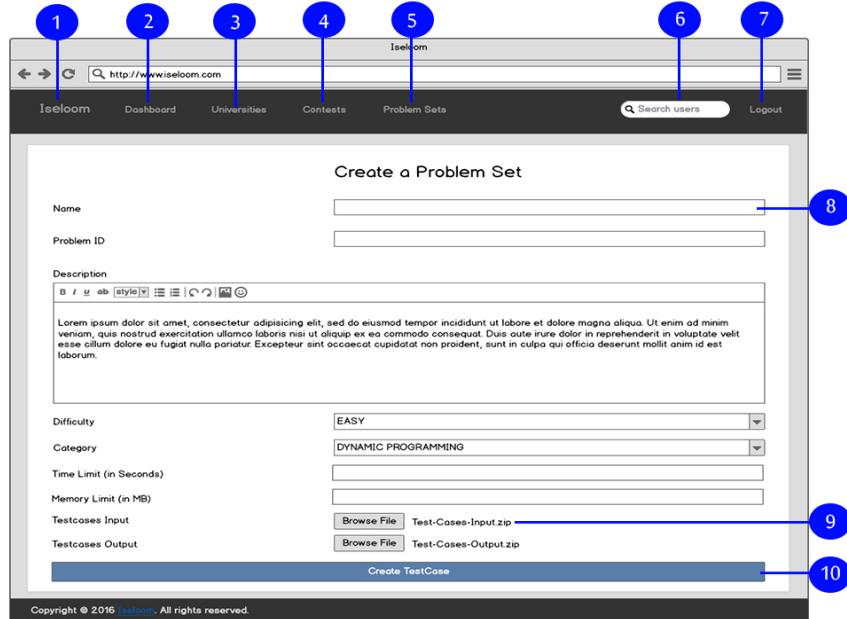
Gambar 3.16 Perancangan Tampilan Halaman Problem Sets



#### Keterangan:

1. Logo sistem *Online Judge*
2. Link “Mike” untuk menuju Halaman Profil dengan pengguna bernama Mike
3. Link “Submissions” untuk menuju Halaman *submissions* pengguna
4. Link “Problem Sets” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
8. Link “Logout” untuk keluar sebagai pengguna dari sistem
9. Deskripsi soal yang akan diselesaikan
10. Button “Choose File” untuk memilih *file* kode program yang akan dikirim
11. Button “Send” untuk mengirim *file* kode program
12. *TextArea* yang dapat diisi kode program
13. Button “Send” untuk mengirim kode program dari *TextArea*

Gambar 3.17 Perancangan Tampilan Halaman Solve Problem



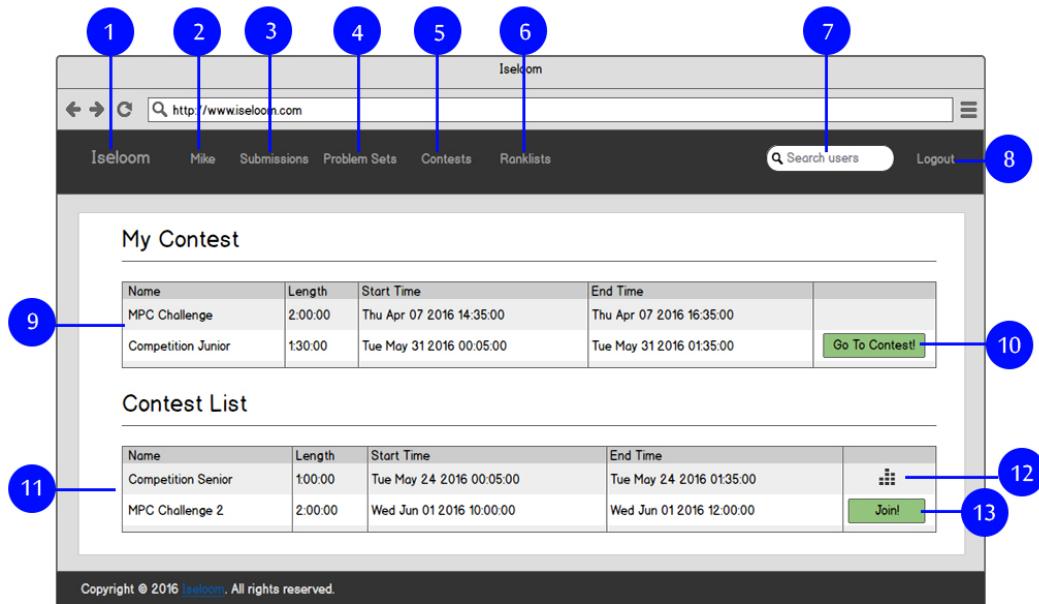
Keterangan:

1. Logo sistem *Online Judge*
2. *Link* “Dashboard” untuk menuju Halaman *Dashboard User*
3. *Link* “Universities” untuk menuju Halaman daftar universitas
4. *Link* “Contests” untuk menuju Halaman daftar kontes
5. *Link* “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. *Link* “Logout” untuk keluar sebagai pengguna dari sistem
8. Data soal yang ingin dibuat
9. *Upload file input* dan *output test case* soal
10. *Button* “Create” untuk membuat soal

Gambar 3.18 Perancangan Tampilan Halaman Create Problem

#### H. Perancangan Tampilan Halaman Contest Member

Pada Gambar 3.19, *member* dapat mengikuti kontes yang sedang berjalan, mendaftar pada kontes yang akan diadakan serta melihat *scoreboard* dan hasil kontes yang ada.



Keterangan:

1. Logo sistem *Online Judge*
2. Link “Mike” untuk menuju Halaman Profil dengan pengguna bernama Mike
3. Link “Submissions” untuk menuju Halaman *submissions* pengguna
4. Link “Problem Sets” untuk menuju Halaman daftar soal
5. Link “Contests” untuk menuju Halaman daftar kontes
6. Link “Ranklists” untuk menuju Halaman daftar urutan pengguna berdasarkan *rating*
7. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
8. Link “Logout” untuk keluar sebagai pengguna dari sistem
9. Daftar kontes yang diikuti pengguna
10. Button “Go To Contest!” untuk memasuki kontes yang telah diikuti pengguna
11. Daftar kontes yang belum diikuti pengguna
12. Link “Scoreboard” untuk melihat *scoreboard* kontes
13. Button “Join!” untuk mengikuti kontes

Gambar 3.19 Perancangan Tampilan Halaman Contest Member

## I. Perancangan Tampilan Halaman Dashboard Admin

Pada Gambar 3.20, admin dapat melihat daftar *member* yang ada dan menverifikasi *member*. Admin juga mampu membuat admin dari *member* yang ada ataupun membuat admin baru.

The screenshot shows the Iseloom Admin Dashboard. At the top, there are links for Dashboard, Universities, Contests, Problem Sets, and a search bar labeled 'Search users'. A 'Logout' link is also present. Below the header, there are two main sections: 'Users' and 'Admin'.

**Users Section:**

No	Name	UserName	Email	University	Date Joined	Activation	Verification	Action Buttons
1	mike	m	m@k.co	Mikroskil	Thu Mar 24 2016 19:55:56 GMT+0700 (WIB)	Activated	Verified	<button>Make as Admin</button>
2	kelvin	k	k@a.co	Binus University	Fri Mar 25 2016 10:12:15 GMT+0700 (WIB)	Activated	Verified	<button>Make as Admin</button>
3	jt	j	j@t.co	Universitas Sumatera Utara	Fri Mar 25 2016 15:15:02 GMT+0700 (WIB)	Not Activated	Verified	<button>Verify</button>

**Admin Section:**

No	Name	UserName	Email	Date Joined	Status
1	jele	jt	je@te.co	Thu Mar 24 2016 19:55:56 GMT+0700 (WIB)	Verified

At the bottom left, there is a button labeled 'Create New Admin'. The footer contains the copyright notice: 'Copyright © 2016 Iseloom. All rights reserved.'

Blue numbered circles from 1 to 12 point to specific elements in the interface:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. SearchBox untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Daftar pengguna yang telah terdaftar dalam sistem
9. Button “Make as Admin” untuk membuat pengguna menjadi admin
10. Button “Verify” untuk menverifikasi pengguna
11. Button “Create New Admin” untuk membuat admin baru
12. Daftar admin pada sistem

Gambar 3.20 Perancangan Tampilan Halaman Dashboard Admin

## J. Perancangan Tampilan Halaman Universities Admin

Pada Gambar 3.21, terdapat daftar universitas yang ada yang digunakan untuk pendaftaran *member*. Admin dapat menambah universitas baru sehingga dapat digunakan oleh *member*.

The screenshot shows a web application interface for managing universities. At the top, there is a navigation bar with links for Iseloom, Dashboard, Universities, Contests, Problem Sets, a search bar labeled 'Search users', and a Logout link. The main content area is titled 'Universities' and contains a table with three rows of data:

No	Name	City	Country
1	STMIK Mikroskil	Medan	Indonesia
2	Binus University	Jakarta	Indonesia
3	Universitas Sumatera Utara	Medan	Indonesia

Below the table, there is a section titled 'Add University' with three input fields: 'University Name', 'City', and 'Country'. A blue button labeled 'Add University' is located at the bottom of this section. The footer of the page includes the text 'Copyright © 2016 Iseloom. All rights reserved.'

Numbered callouts point to specific elements:

- 1: Logo in the top left.
- 2: 'Dashboard' link in the top menu.
- 3: 'Universities' link in the top menu.
- 4: 'Contests' link in the top menu.
- 5: 'Problem Sets' link in the top menu.
- 6: Search bar in the top right.
- 7: 'Logout' link in the top right.
- 8: 'Universities' table.
- 9: 'Add University' input field.
- 10: 'Add University' button.

Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. TextBox untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Daftar universitas pada sistem
9. TextBox yang diisi untuk menambah universitas baru
10. Button “Add University” untuk menambah universitas baru

Gambar 3.21 Perancangan Tampilan Halaman Universities Admin

## K. Perancangan Tampilan Halaman Contest Admin

Pada Gambar 3.22, terdapat daftar kontes yang ada serta waktu kontes diadakan. Pada Gambar 3.23, admin dapat membuat kontes baru yang dapat diikuti *member*. Pada Gambar 3.24, admin dapat mengubah data kontes serta membatalkan kontes yang selesai. Pada Gambar 3.25, admin dapat menambah soal pemrograman yang perlu diselesaikan *member* pada kontes dengan urutan soal. Admin juga dapat melihat kontestan yang sudah mendaftar pada kontes. Pada Gambar 3.26, *scoreboard* kontes ditampilkan dengan jelas baik peserta kontes, soal pada kontes hingga *ranking* pada kontes. Pada Gambar 3.27, admin dapat melihat semua *submission* yang dilakukan oleh peserta kontes dan hasil dari *submission* peserta. Pada Gambar 3.28, admin dapat melihat secara detil *submission* peserta kontes dari kode program hingga hasil dalam pengujian kode program.

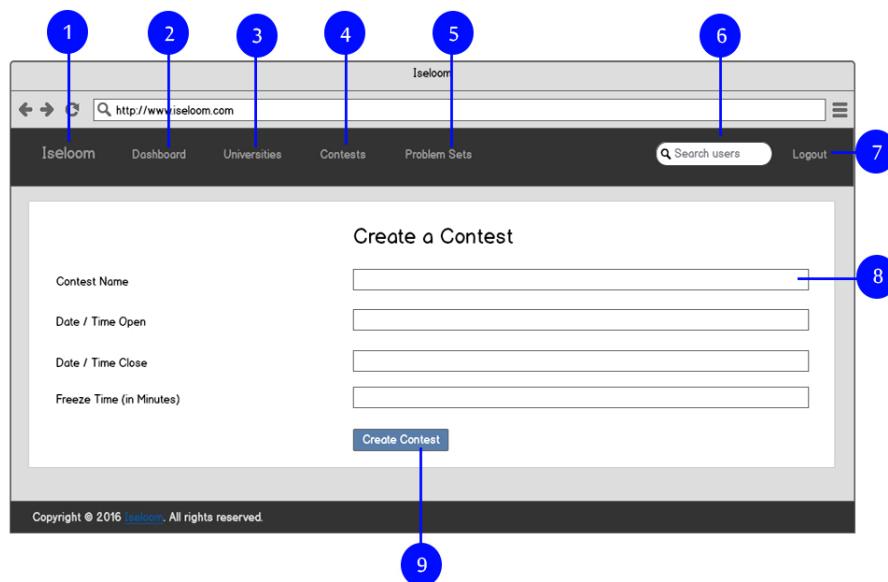
Name	Length	Start Time	End Time	
MPC Challenge	100:00	Thu Feb 25 2016 11:00:00	Sat Jun 18 2016 11:00:00	<span>Edit</span> <span>Delete</span>
Mikroskil Challenge	8:00:00	Wed Mar 23 2016 20:41:00	Wed Mar 23 2016 22:00:00	<span>Edit</span> <span>Delete</span>

Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem

8. Daftar kontes pada *online judge*
9. *Button* “Scoreboard” untuk melihat *scoreboard* kontes
10. *Button* “Problems” untuk menambah soal pada kontes
11. *Button* “Edit Contest” untuk mengubah data kontes
12. *Button* “Delete Contest” untuk menghapus kontes

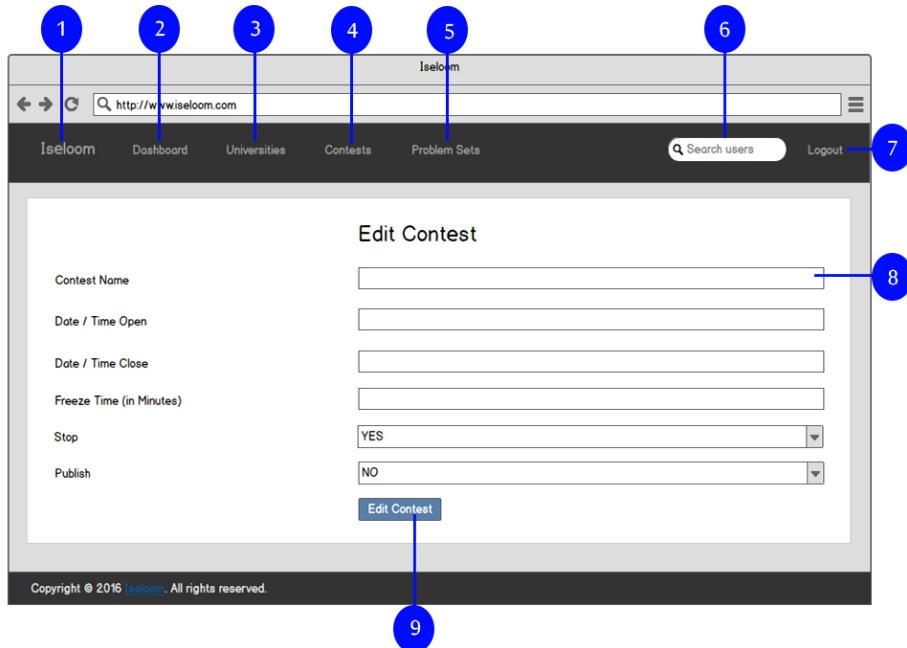
Gambar 3.22 Perancangan Tampilan Halaman Contest Admin



Keterangan:

1. Logo sistem *Online Judge*
2. *Link* “Dashboard” untuk menuju Halaman *Dashboard User*
3. *Link* “Universities” untuk menuju Halaman daftar universitas
4. *Link* “Contests” untuk menuju Halaman daftar kontes
5. *Link* “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. *Link* “Logout” untuk keluar sebagai pengguna dari sistem
8. *TextBox* data kontes yang akan dibuat
9. *Button* “Create Contest” untuk membuat kontes baru

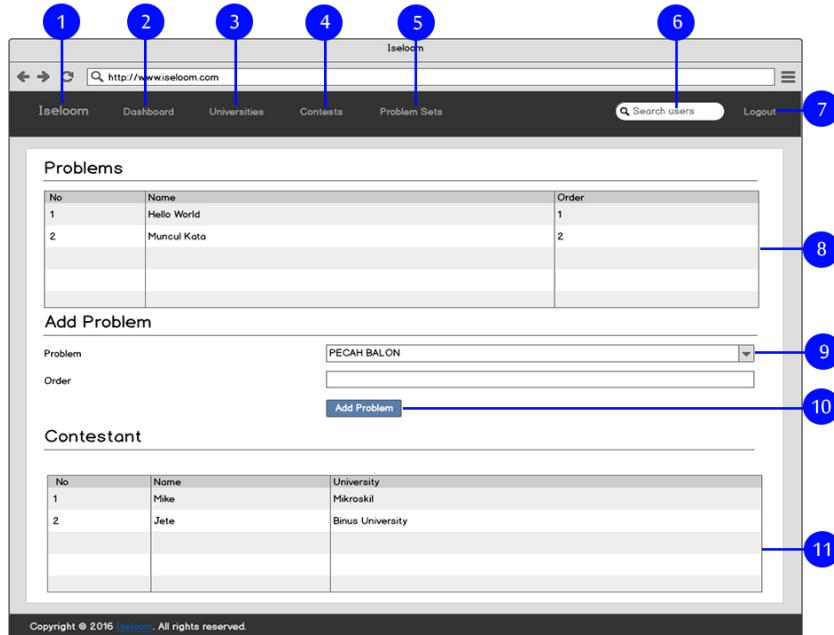
Gambar 3.23 Perancangan Tampilan Halaman Create Contest Admin



Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. *TextBox* data kontes yang ingin diubah
9. *Button* “Edit Contest” untuk mengubah data kontes

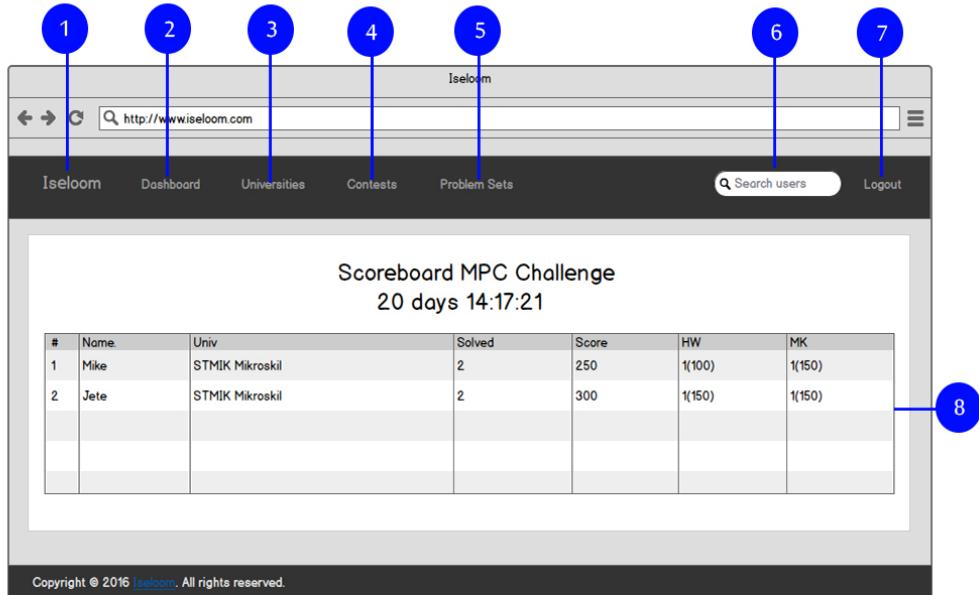
Gambar 3.24 Perancangan Tampilan Halaman Edit Contest Admin



#### Keterangan:

1. Logo sistem *Online Judge*
2. *Link* “Dashboard” untuk menuju Halaman *Dashboard User*
3. *Link* “Universities” untuk menuju Halaman daftar universitas
4. *Link* “Contests” untuk menuju Halaman daftar kontes
5. *Link* “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. *Link* “Logout” untuk keluar sebagai pengguna dari sistem
8. Daftar soal pada kontes
9. Data soal yang ingin ditambahkan pada kontes
10. *Button* “Add Problem” untuk menambah soal kedalam kontes
11. Daftar kontestan yang telah mendaftar dalam kontes

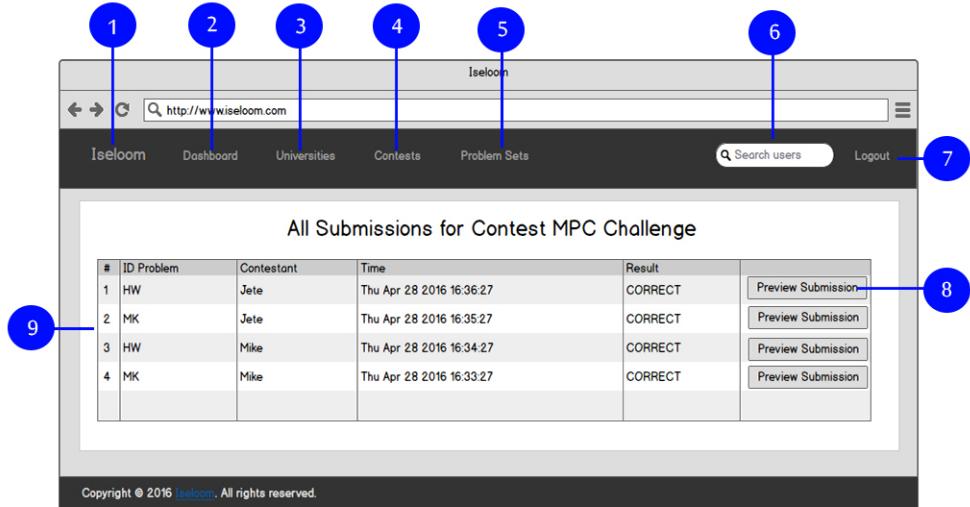
Gambar 3.25 Perancangan Tampilan Halaman Problem Contest



Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Table *scoreboard* pada kontes

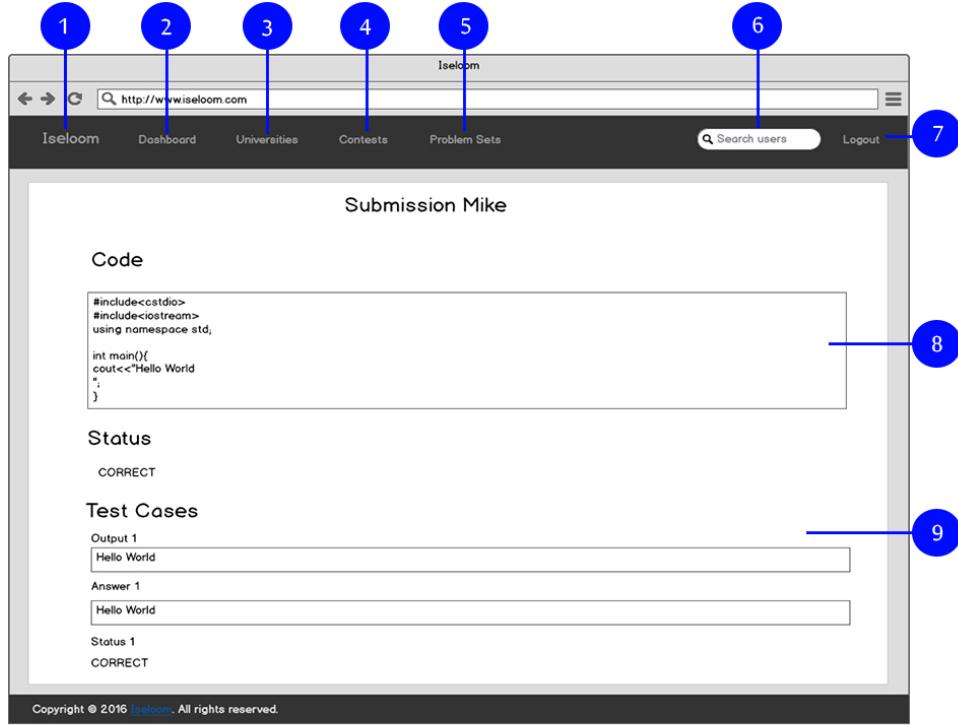
Gambar 3.26 Perancangan Tampilan Halaman Scoreboard Contest



Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Button “Preview Submission” untuk melihat detail *submission*
9. Tabel daftar *submissions* pada kontes

Gambar 3.27 Perancangan Tampilan Halaman Submissions Contest



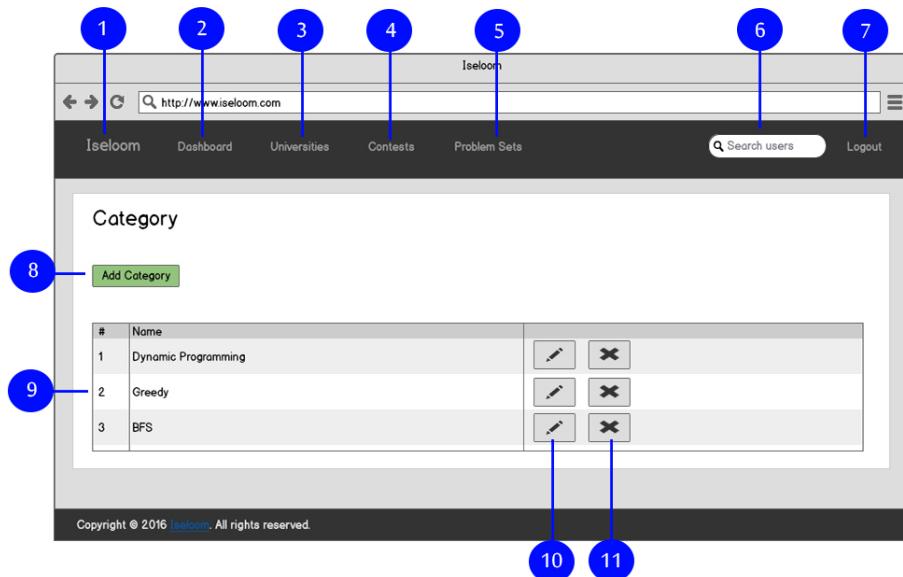
#### Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Kode program yang dikirim oleh pengguna
9. Pengujian kode program dengan *test cases*

Gambar 3.28 Perancangan Tampilan Halaman Submission Detail Contest

## L. Perancangan Tampilan Halaman Problem Category

Pada Gambar 3.29, admin dapat menambah kategori soal pemrograman yang dapat digunakan untuk men-*filter* soal pemrograman. Admin juga dapat mengubah atau menghapus kategori yang ada.



Keterangan:

1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. *SearchBox* untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Button “Add Category” untuk menambah kategori soal
9. Tabel daftar kategori soal yang ada
10. Button “Edit” untuk mengubah kategori
11. Button “Remove” untuk menghapus kategori

Gambar 3.29 Perancangan Tampilan Halaman Problem Category

## M. Perancangan Tampilan Halaman Problem Sets Admin

Pada Gambar 3.30, admin dapat melihat soal pemrograman yang ada baik yang telah ter-*publish* ataupun yang belum. Soal pemrograman yang telah ter-*publish* akan dapat diselesaikan oleh *member* secara bebas, sedangkan soal yang belum ter-*publish* dapat dengan aman digunakan pada kontes karena belum dapat diselesaikan oleh *member*. Admin juga dapat mengubah atau menghapus soal pemrograman yang ada.



Keterangan:

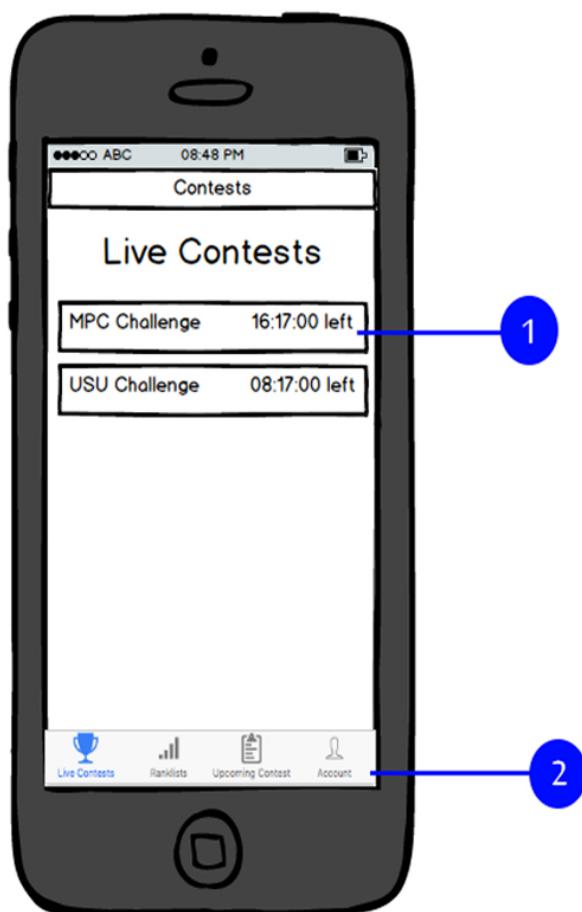
1. Logo sistem *Online Judge*
2. Link “Dashboard” untuk menuju Halaman *Dashboard User*
3. Link “Universities” untuk menuju Halaman daftar universitas
4. Link “Contests” untuk menuju Halaman daftar kontes
5. Link “Problem Sets” untuk menuju Halaman daftar soal
6. SearchBox untuk melakukan pencarian pengguna berdasarkan nama
7. Link “Logout” untuk keluar sebagai pengguna dari sistem
8. Tabel daftar soal yang sudah di-*publish*
9. Button “Edit” untuk mengubah data soal
10. Button “Remove” untuk menghapus soal
11. Button “Unpublish” untuk tidak mem-*publish* soal

12. Tabel daftar soal yang belum di-*publish*
13. *Button* “Preview” untuk mencoba soal untuk diuji oleh admin
14. *Button* “Publish” untuk mem-*publish* soal

Gambar 3.30 Perancangan Tampilan Halaman Problem Sets Admin

N. Perancangan Tampilan Mobile Live Contest

Pada Gambar 3.31, pengguna dapat melihat daftar kontes yang sedang berlangsung. Pengguna memilih kontes untuk melihat detail *scoreboard* dari kontes tersebut.



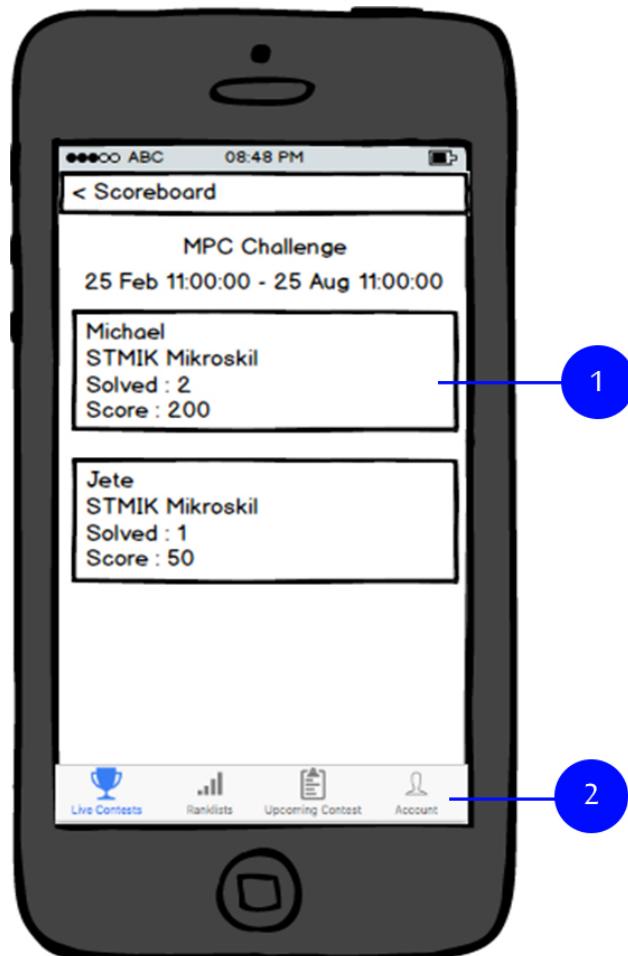
Keterangan:

1. Daftar kontes yang sedang berlangsung
2. Menu pada aplikasi *mobile*

Gambar 3.31 Perancangan Tampilan Mobile Live Contest

O. Perancangan Tampilan Mobile Scoreboard

Pada Gambar 3.32, pengguna dapat melihat daftar *scoreboard* dari kontes yang dipilih. Pengguna memilih kontestan untuk melihat detail dari soal yang sudah kontestan selesaikan atau belum.



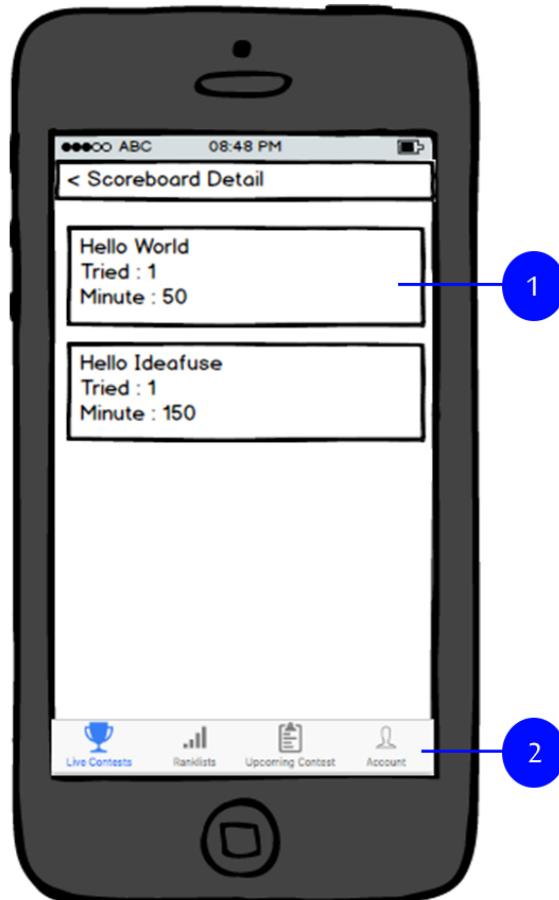
Keterangan:

1. Data *ranking* pada kontes berdasarkan *scoreboard*
2. Menu pada aplikasi *mobile*

Gambar 3.32 Perancangan Tampilan Mobile Scoreboard

P. Perancangan Tampilan Mobile Scoreboard Detail

Pada Gambar 3.33, pengguna dapat melihat daftar soal yang sudah diselesaikan atau belum dari kontestan yang dipilih.



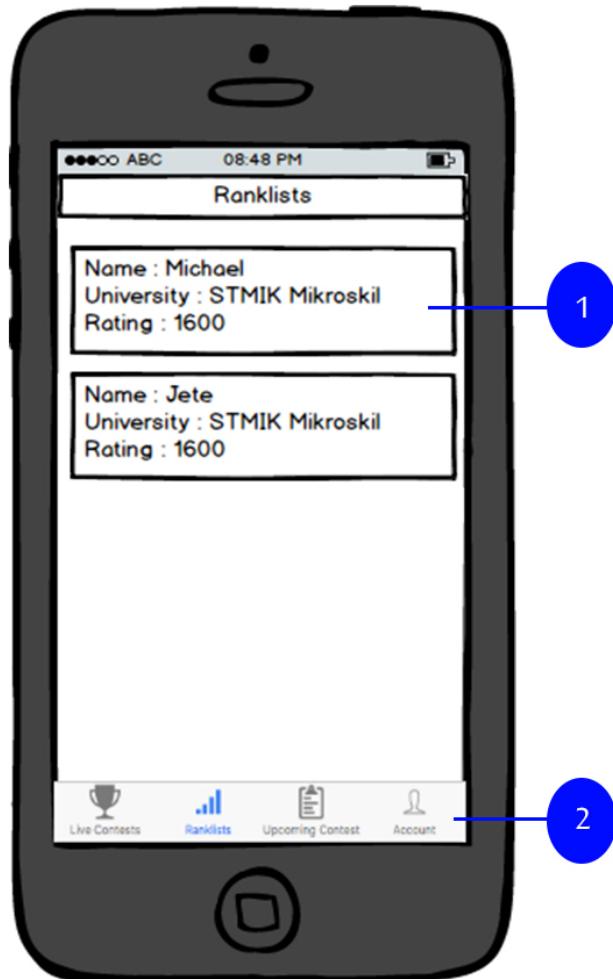
Keterangan:

1. Detil dari soal pada kontes berdasarkan pengguna
2. Menu pada aplikasi *mobile*

Gambar 3.33 Perancangan Tampilan Mobile Scoreboard Detail

Q. Perancangan Tampilan Mobile Ranklist

Pada Gambar 3.34, pengguna bisa melihat urutan daftar *member* berdasarkan urutan *rating*.



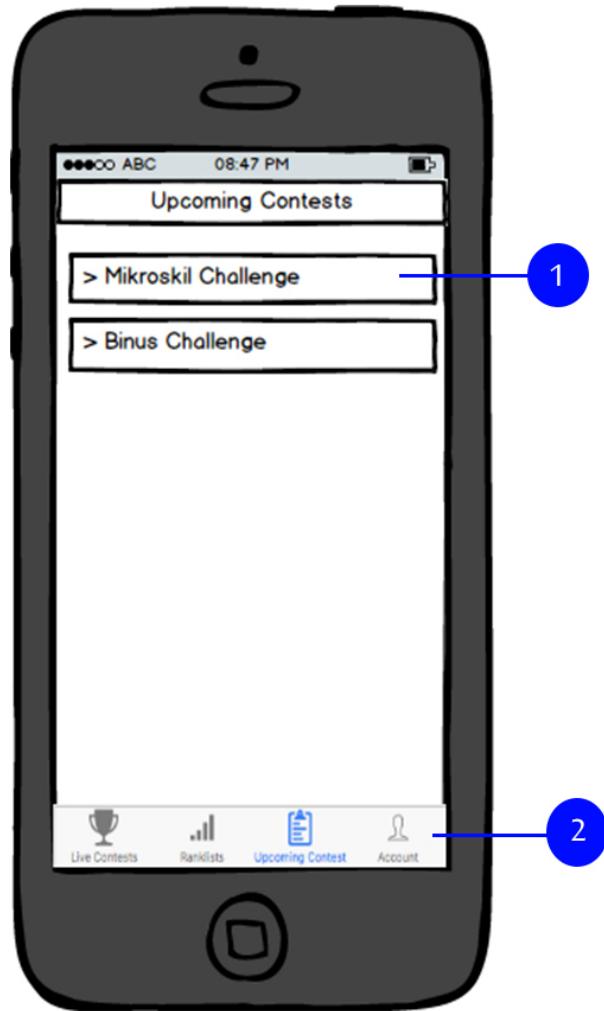
Keterangan:

1. Daftar *ranklists* pengguna berdasarkan urutan *rating*
2. Menu pada aplikasi *mobile*

Gambar 3.34 Perancangan Tampilan Mobile Ranklist

R. Perancangan Tampilan Mobile Upcoming Contest

Pada Gambar 3.35, pengguna dapat melihat daftar kontes yang akan dilaksanakan. Pengguna memilih kontes untuk melihat detail dari kontes yang akan diikuti atau tidak.



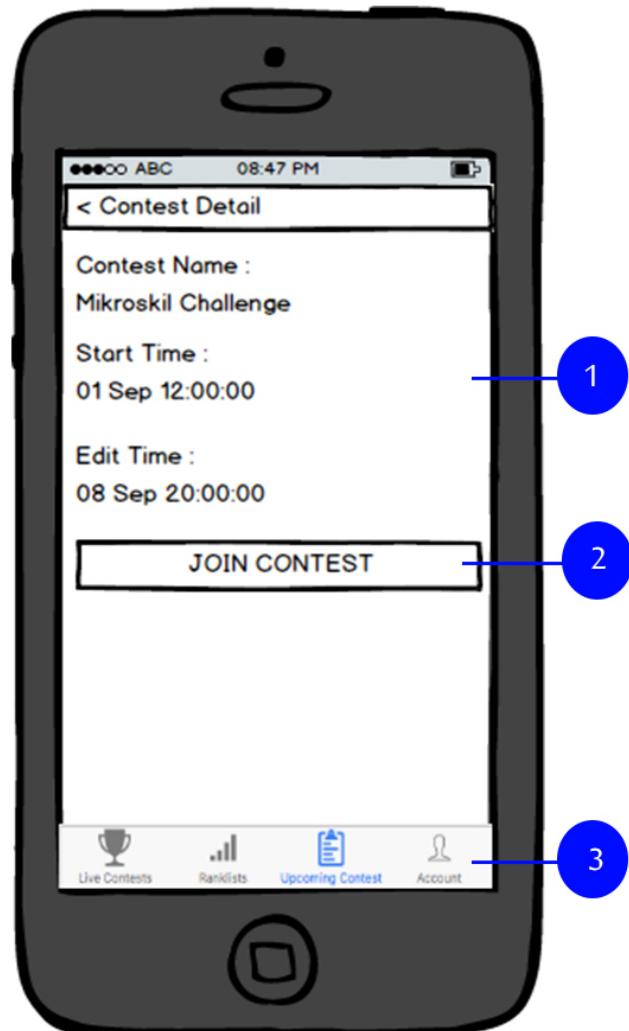
Keterangan:

1. Daftar kontes yang akan diadakan pada *online judge*
2. Menu pada aplikasi *mobile*

Gambar 3.35 Perancangan Tampilan Mobile Upcoming Contest

S. Perancangan Tampilan Mobile Upcoming Contest Detail

Pada Gambar 3.36, pengguna dapat melihat detail dari kontes yang akan dimulai. Pengguna dapat memilih untuk mengikuti kontes atau tidak mengikuti kontes.



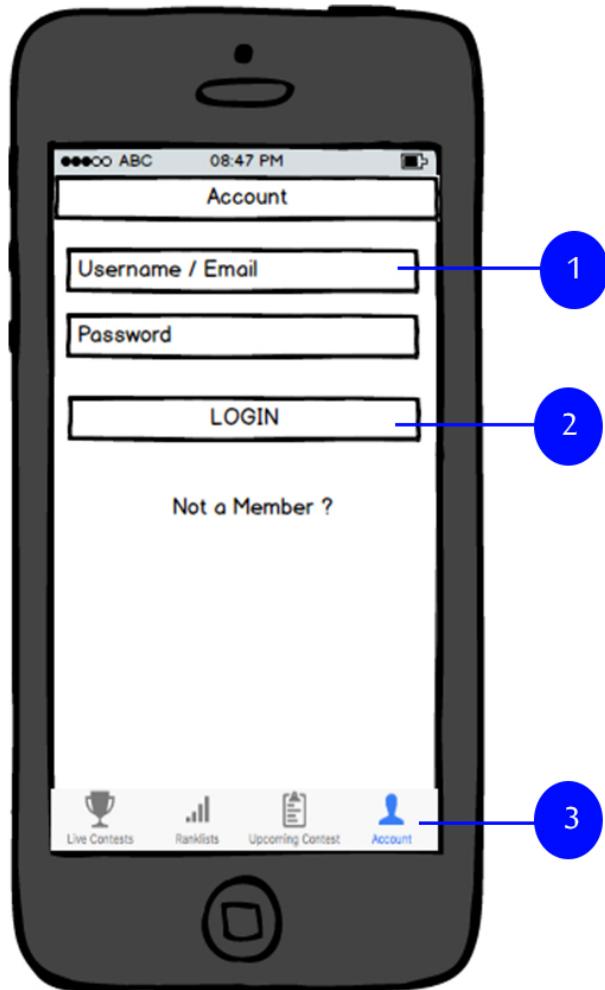
Keterangan:

1. Data detail dari kontes yang akan diadakan
2. Button “Join Contest” untuk mengikuti kontes
3. Menu pada aplikasi *mobile*

Gambar 3.36 Perancangan Tampilan Mobile Upcoming Contest Detail

### T. Perancangan Tampilan Mobile Login

Pada Gambar 3.37, pengguna dapat masuk ke dalam sistem apabila memiliki keanggotaan sebagai *member*.



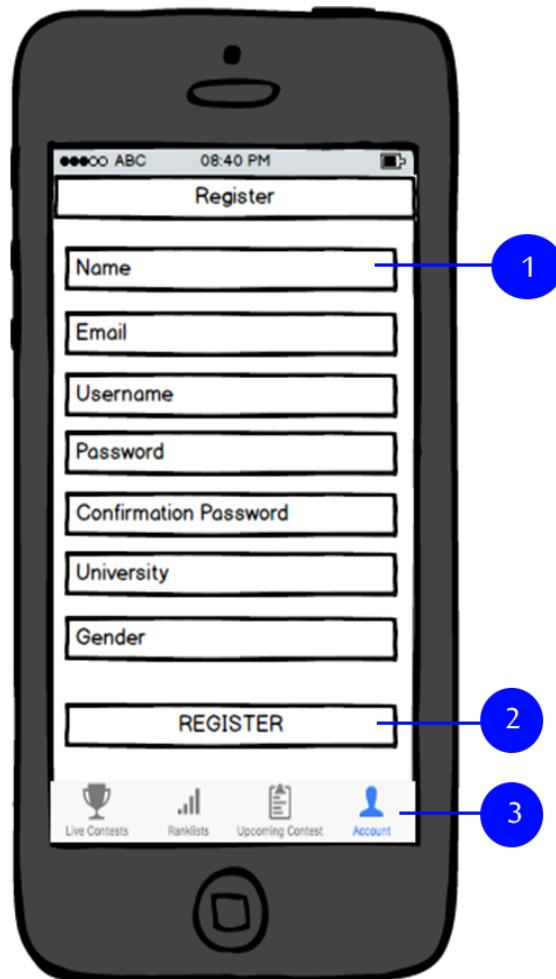
Keterangan:

1. Data dari pengguna untuk masuk ke dalam sistem
2. Button “Login” untuk masuk kedalam sistem
3. Menu pada aplikasi *mobile*

Gambar 3.37 Perancangan Tampilan Mobile Login

#### U. Perancangan Tampilan Mobile Register

Pada Gambar 3.38, pengguna dapat mendaftar diri pada aplikasi untuk dapat menggunakan aplikasi sebagai *member*.



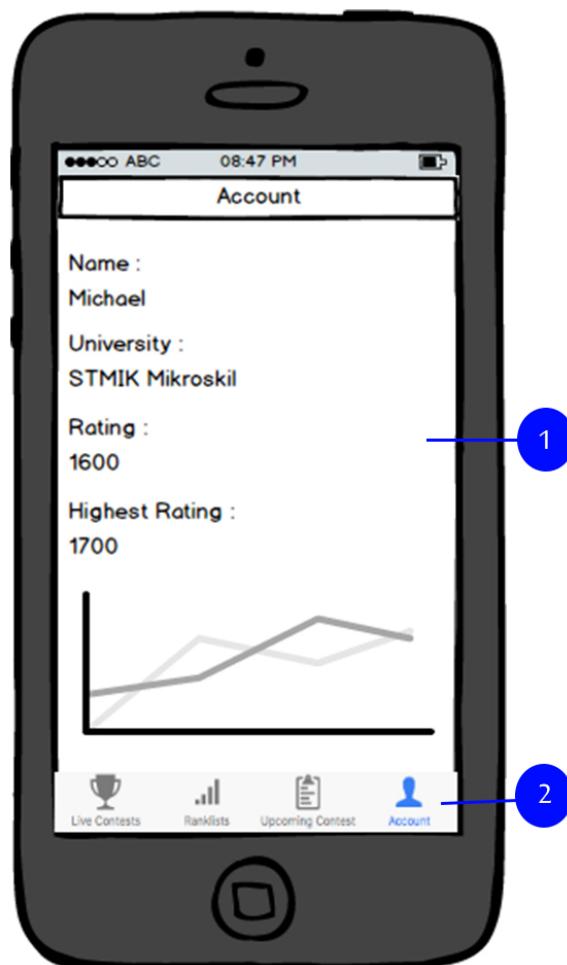
Keterangan:

1. Data pengguna yang ingin mendaftar kedalam sistem
2. Button “Register” untuk daftar kedalam sistem
3. Menu pada aplikasi *mobile*

Gambar 3.38 Perancangan Tampilan Mobile Register

## V. Perancangan Tampilan Mobile Account

Pada Gambar 3.39, pengguna dapat melihat detil dari akun pengguna sendiri.



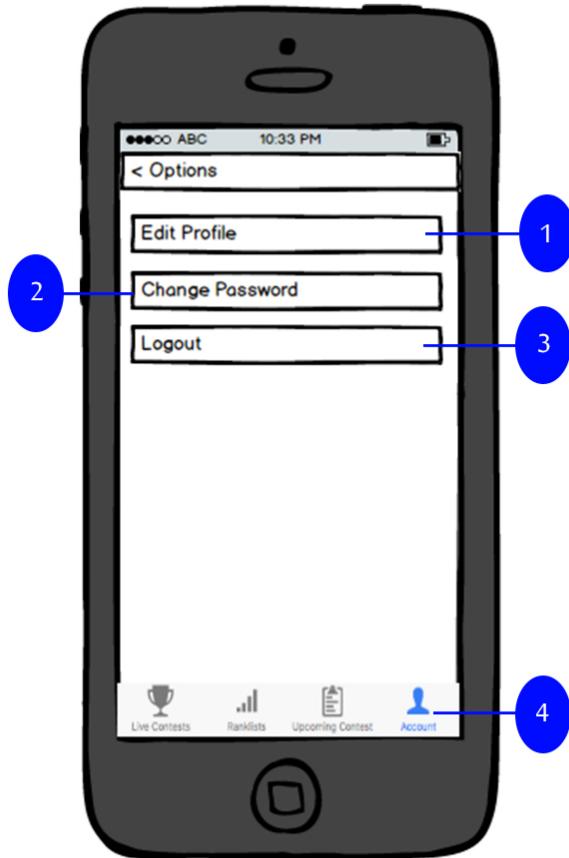
Keterangan:

1. Data pengguna dan grafik *rating* pengguna
2. Menu pada aplikasi *mobile*

Gambar 3.39 Perancangan Tampilan Mobile Account

## W. Perancangan Tampilan Mobile Options

Pada Gambar 3.40, pengguna dapat memilih untuk mengedit profil, mengubah *password* atau *logout* dari akun pada aplikasi.



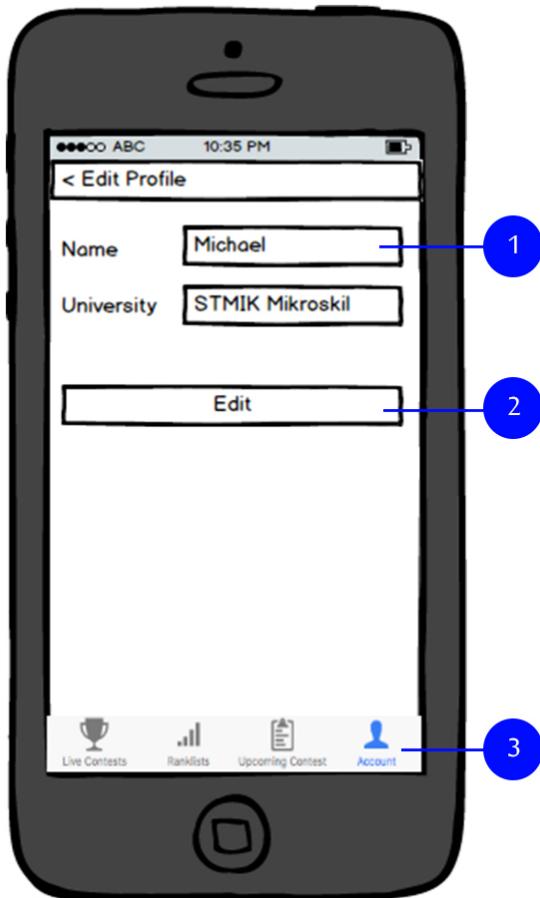
Keterangan:

1. *Button* “Edit Profile” untuk mengubah profil pengguna
2. *Button* “Change Password” untuk mengubah kata sandi pengguna
3. *Button* “Logout” untuk keluar dari sistem
4. Menu pada aplikasi *mobile*

Gambar 3.40 Perancangan Tampilan Mobile Options

## X. Perancangan Tampilan Mobile Edit Profile

Pada Gambar 3.41, pengguna dapat mengedit profil dari akun pengguna sendiri.



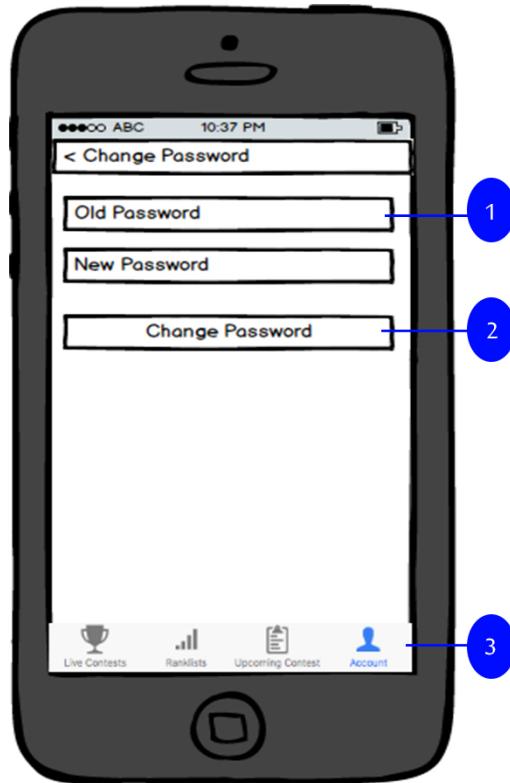
Keterangan:

1. *TextBox* data pengguna yang ingin diubah
2. *Button* “Edit” untuk mengubah data pengguna
3. Menu pada aplikasi *mobile*

Gambar 3.41 Perancangan Tampilan Mobile Edit Profile

#### Y. Perancangan Tampilan Mobile Change Password

Pada Gambar 3.42, pengguna dapat mengubah password dari akun pengguna sendiri.



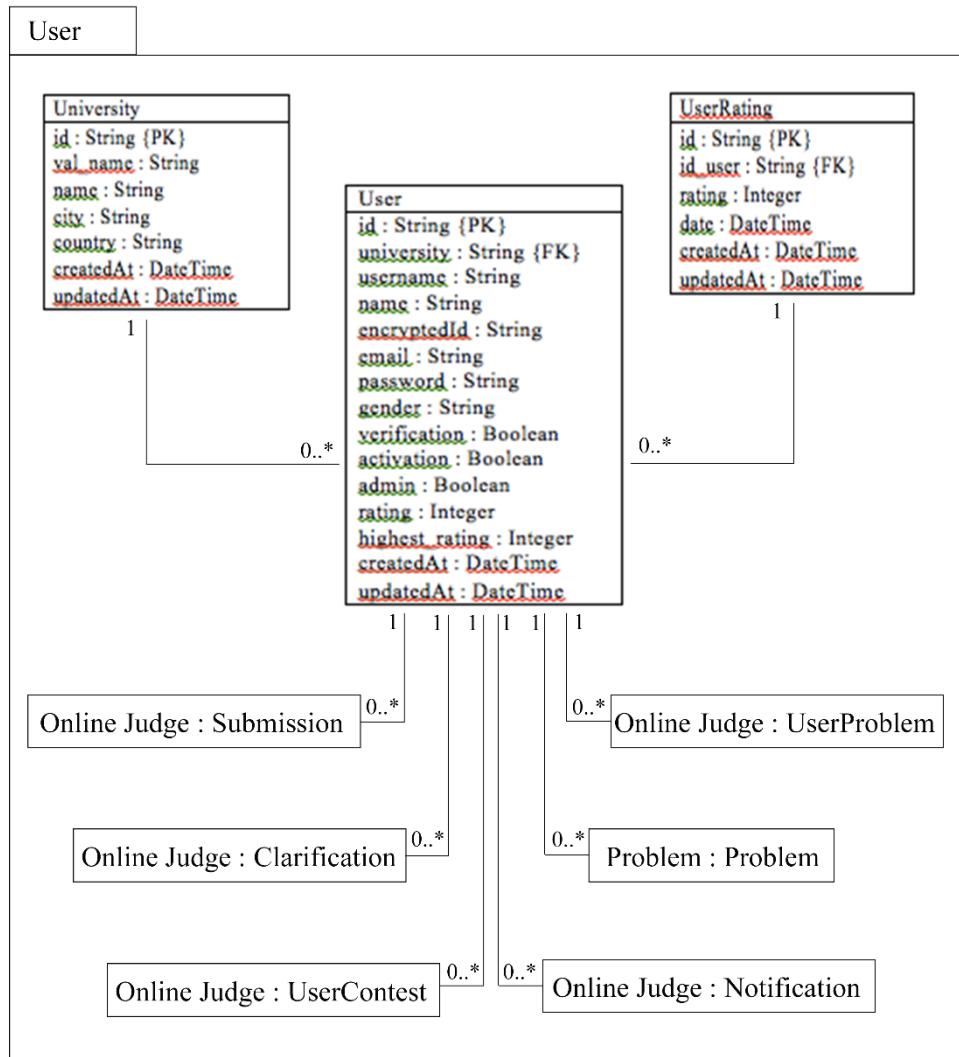
Keterangan:

1. *TextBox* kata sandi pengguna
2. *Button* “Change Password” untuk mengubah kata sandi pengguna
3. Menu pada aplikasi *mobile*

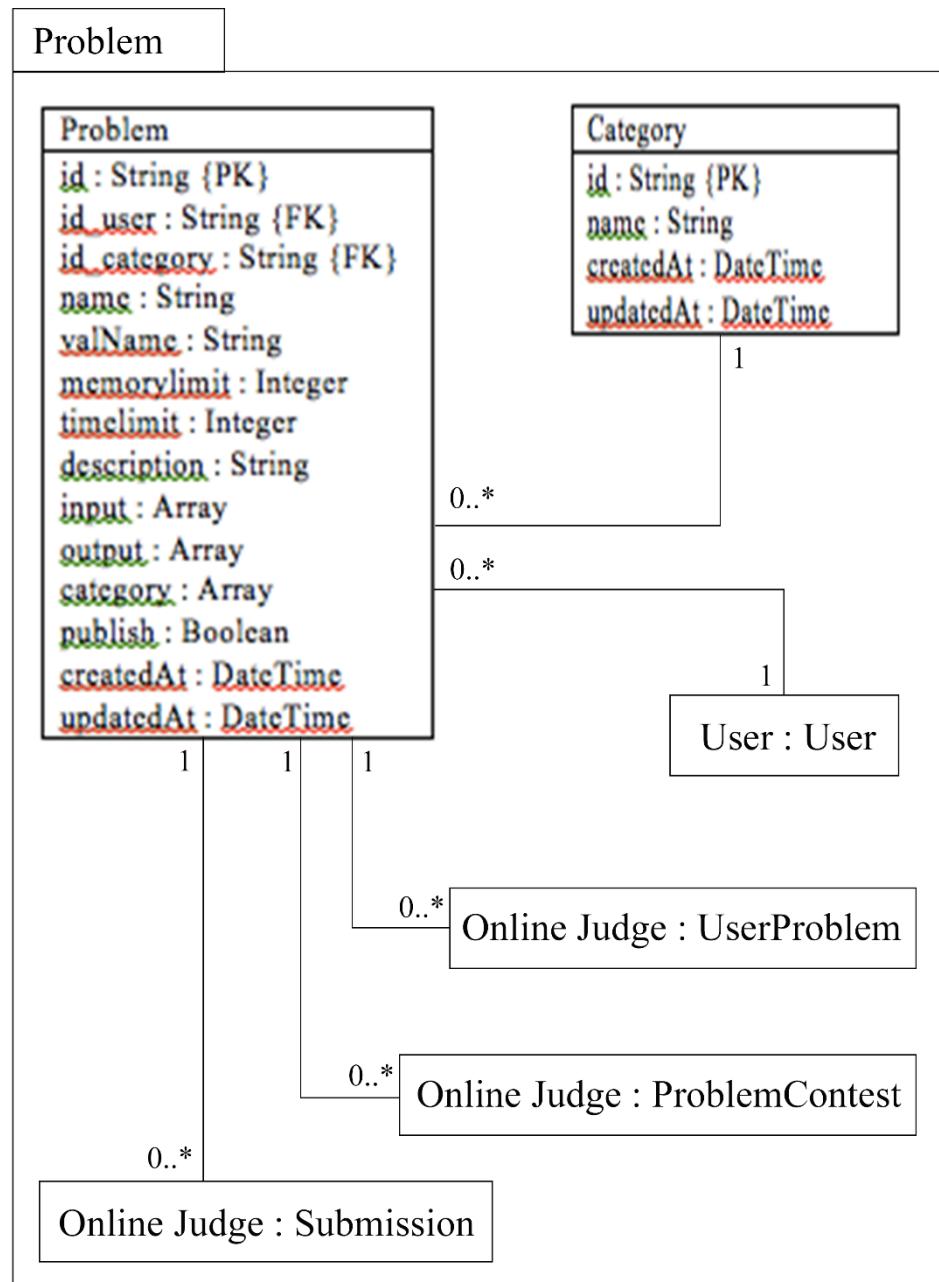
Gambar 3.42 Perancangan Tampilan Mobile Change Password

#### 3.2.3 Perancangan Basis Data

Sistem *online judge* dibuat dengan menggunakan bahasa pemrograman NodeJs serta menggunakan basis data MongoDB. Dalam perancangan basis data, dapat dilihat berdasarkan *class diagram* yang telah dipecah dengan menggunakan UML Package pada Gambar 3.43, Gambar 3.44 dan Gambar 3.45.

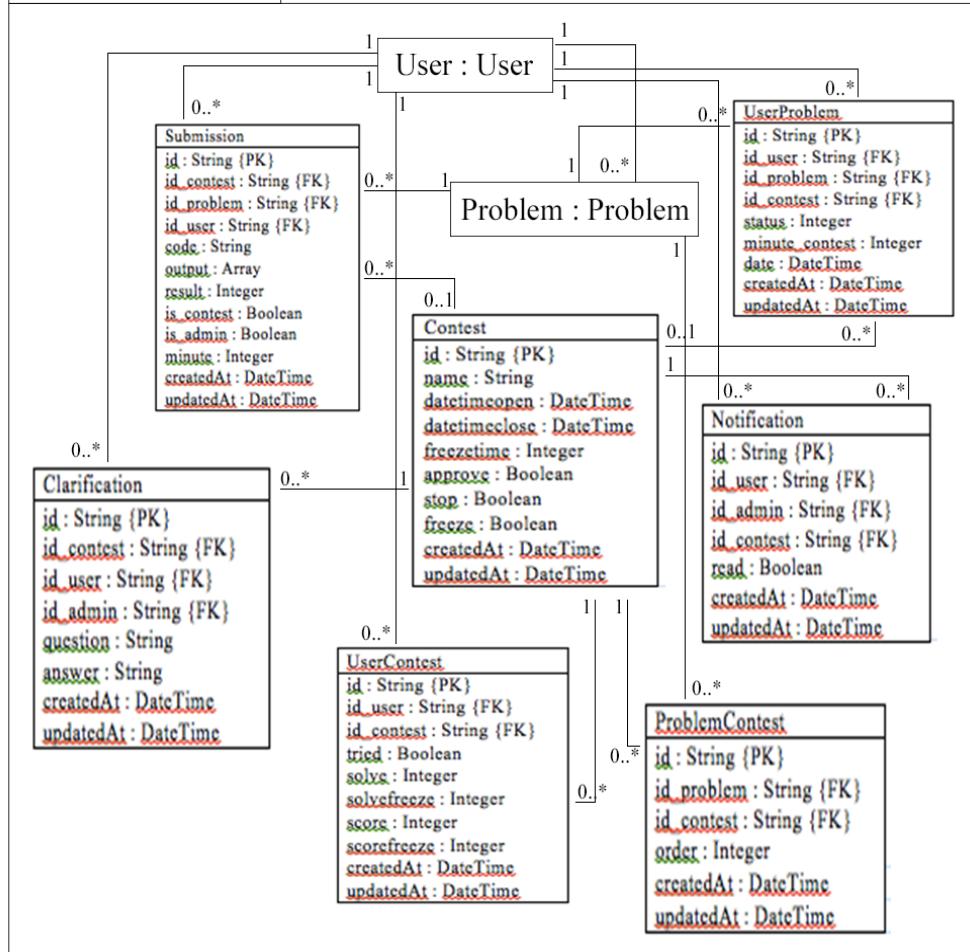


Gambar 3.43 Class Diagram User Package Basis Data



Gambar 3.44 Class Diagram Problem Package Basis Data

## Online Judge



Gambar 3.45 Class Diagram Online Judge Package Basis Data

## **BAB IV**

### **HASIL DAN IMPLEMENTASI**

#### **4.1 Hasil**

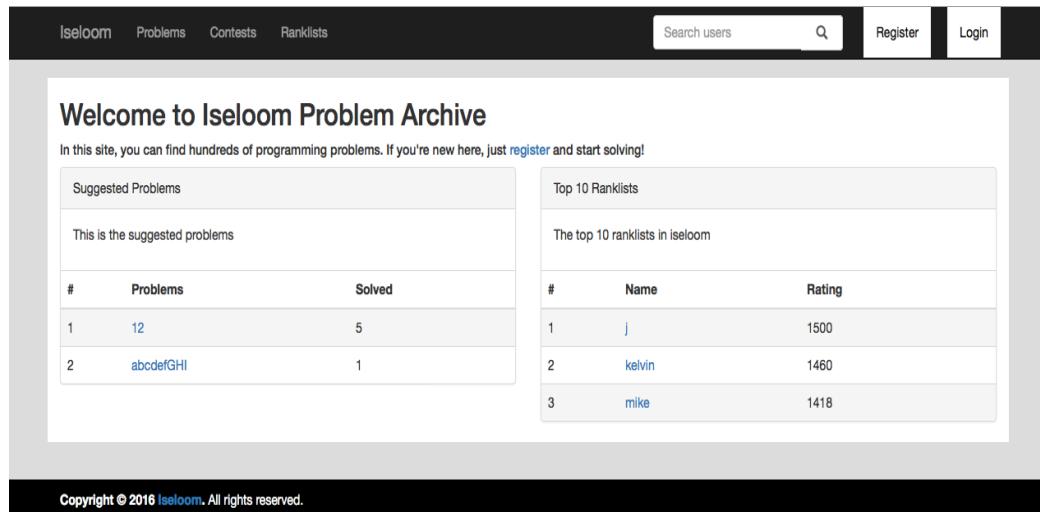
Untuk menggunakan atau menjalankan sistem *online judge* yang dirancang, dibutuhkan *browser* seperti Safari, Google Chrome, dkk. Berikut tampilan hasil aplikasi *online judge* dan *tracking* pelajar yaitu sebagai berikut:

##### 1. Tampilan *Website*

Tampilan *Website* terdiri dari beberapa halaman yaitu sebagai berikut:

###### a. Tampilan Halaman Depan Aplikasi

Tampilan halaman depan aplikasi yang berisi soal pemrograman yang populer dan pengguna yang memiliki *rating* tertinggi dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan Halaman Depan Aplikasi

###### b. Tampilan Halaman Login

Tampilan halaman *login* dimana pengguna masuk ke dalam aplikasi untuk menggunakan aplikasi *Online Judge*. Pengguna wajib mengisi *email* atau *username* serta *password* seperti Gambar 4.2.

The screenshot shows the Iseloom website's login interface. At the top, there is a dark navigation bar with links for 'Iseloom', 'Problems', 'Contests', and 'Ranklists'. To the right of these are search, register, and login buttons. The main content area has a light gray background and features a 'Login' form. The form includes fields for 'Email or Username' and 'Password', a 'Remember me' checkbox, and a 'Login' button. Below the form is a link to 'Forgot your password?'. At the bottom of the page, a black footer bar contains the copyright notice 'Copyright © 2016 Iseloom. All rights reserved.'

Gambar 4.2 Tampilan Halaman Login

c. Tampilan Halaman Register

Tampilan halaman *register*, dimana pengguna mendaftarkan diri untuk dapat menggunakan aplikasi sebagai *member* yang dapat dilihat pada Gambar 4.3.

The screenshot shows the Iseloom website's registration interface. At the top, there is a dark navigation bar with links for 'Iseloom', 'Problems', 'Contests', and 'Ranklists'. To the right of these are search, register, and login buttons. The main content area has a light gray background and features a 'Register' form. The form includes fields for 'Name', 'Email', 'Username', 'Password', and 'Confirmation Password'. Below these are dropdown menus for 'University' (set to 'STMIK Mikroskil') and 'Gender' (set to 'Male'). A 'Register' button is located at the bottom of the form. At the bottom of the page, a black footer bar contains the copyright notice 'Copyright © 2016 Iseloom. All rights reserved.'

Gambar 4.3 Tampilan Halaman Register

d. Tampilan Halaman Ranklists

Tampilan halaman *ranklists* yang dapat berupa *ranklists* user ataupun univeristas. Urutan dilakukan berdasarkan *rating* tertinggi hingga terendah, untuk universitas digunakan *rating* rata-rata. Tampilan *ranklists* user dapat dilihat pada Gambar 4.4 sedangkan *ranklists* universitas dapat dilihat pada Gambar 4.5.

No.	User	University	Rating
1	Denny Ho	STMIK Mikroskil	1639
2	Fredy Ong jaya	STMIK Mikroskil	1555
3	Benny wijaya	STMIK Mikroskil	1548
4	Erick Kwan	STMIK Mikroskil	1537
5	KX	STMIK Mikroskil	1517
6	Widya Putra	STMIK Mikroskil	1511
7	Tommy	STMIK Mikroskil	1484
8	Prajoko Yuanto	Universitas Indonesia	1481
9	Stephanie	STMIK Mikroskil	1481
10	Kelvin Hermawan Leonardo	STMIK Mikroskil	1475
11	davin	STMIK Mikroskil	1473
11	Ferry Irawan	STMIK Mikroskil	1473

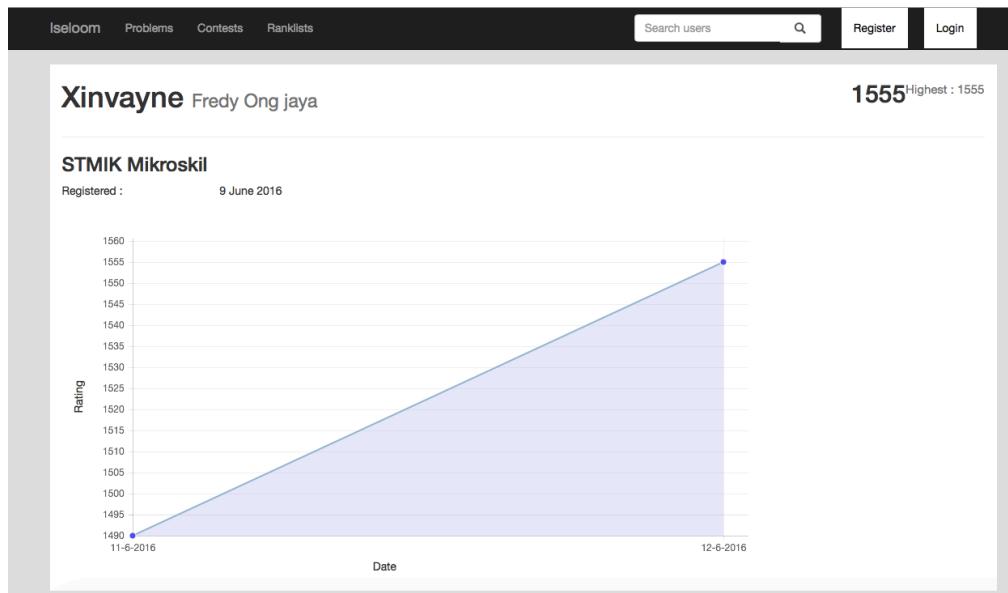
Gambar 4.4 Tampilan Ranklists User

No.	University	Members	Rating
1	STMIK Mikroskil	17	1502
2	Universitas Indonesia	4	1475
3	Binus University	1	1473
3	Universitas Prima Indonesia	2	1473

Gambar 4.5 Tampilan Ranklists Universitas

#### e. Tampilan Halaman Profil

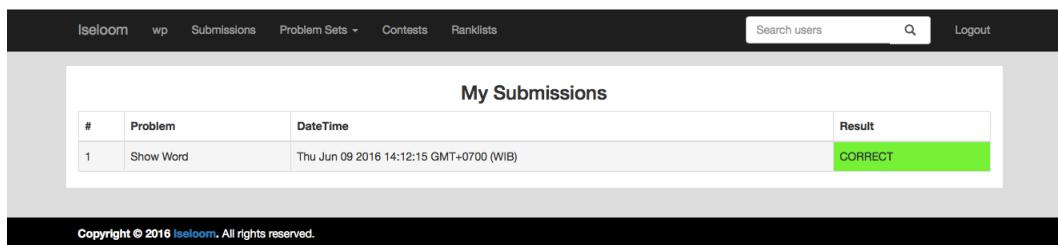
Tampilan halaman profil dari *member* mencakup nama, universitas hingga *rating* dari *user* tersebut. Untuk grafik pergerakan *rating* akan diambil 10 perubahan *rating* terakhir. *Member* yang membuka profil dirinya sendiri mampu mengubah data seperti nama, universitas serta *password*. Tampilan Halaman Profil dapat dilihat pada Gambar 4.6.



Gambar 4.6 Tampilan Profil Member

f. Tampilan Halaman Submission Member

Tampilan halaman *submission* yang pernah dilakukan *member* terhadap soal pemrograman yang bebas diselesaikan diluar kontes. Tampilan dapat dilihat pada Gambar 4.7.



Gambar 4.7 Tampilan Submission Member

g. Tampilan Halaman List Problem Sets Member

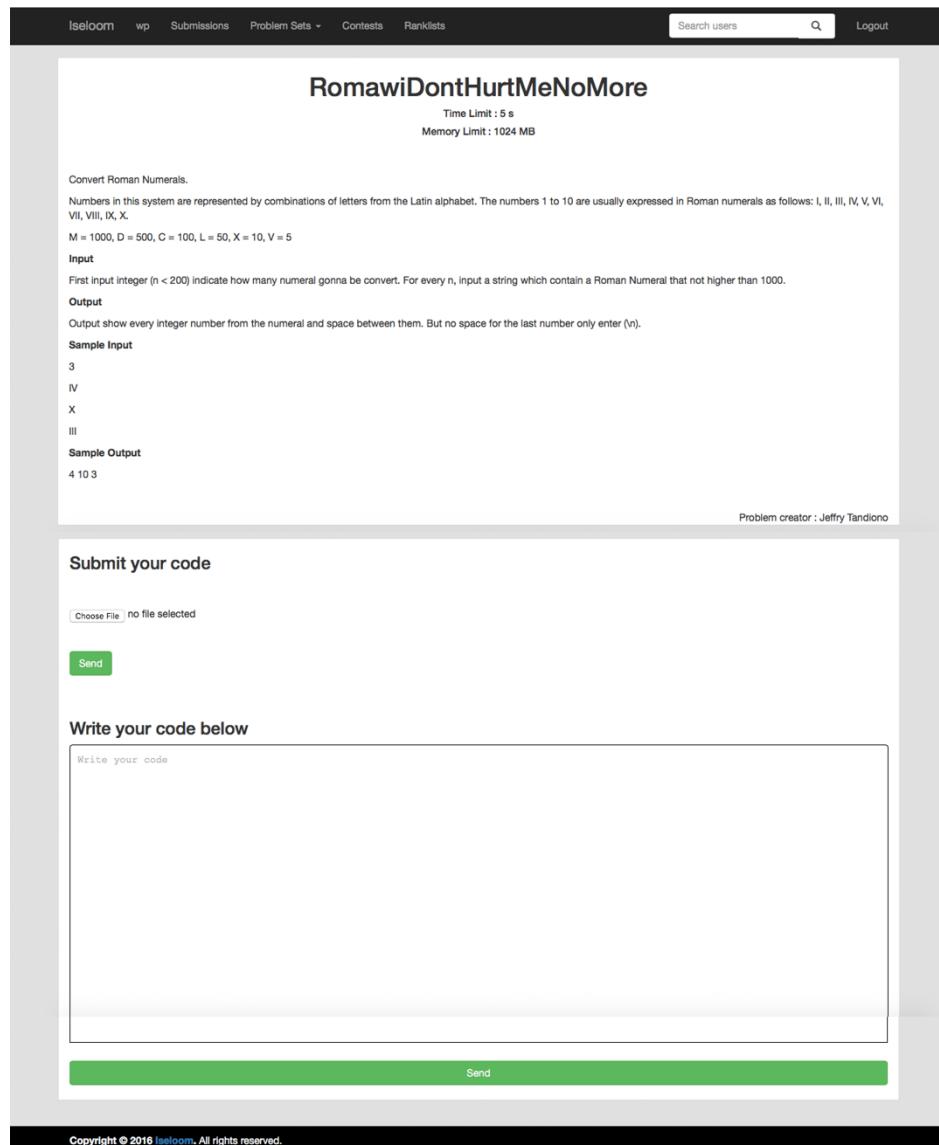
Tampilan halaman *list problem* yang tersedia untuk *member* tanpa mengikuti kontes. *Member* dapat melihat apakah soal pernah diselesaikan atau tidak, tampilan halaman dapat dilihat pada Gambar 4.8.

NAME	SUBMISSIONS			CATEGORY	SOLVED
	TOTAL	ACC.	RATIO		
Hello World	1	1	100%	Simple	⌚
Show Word	22	8	36%	Simple	⌚
Pascal's Triangle	0	0	0%	Simple	⌚
Guessing Game	0	0	0%	Greedy,Simple	⌚
Artechno Lamp	0	0	0%	Simple	⌚
Conversion	1	1	100%	Simple	⌚
Do You Have Enough Money?	1	1	100%	Simple	⌚
Finite Loop	0	0	0%	Dynamic Programming,Greedy,Simple	⌚
Save Mr Oski	0	0	0%	Simple	⌚
RomawiIDontHurtMeNoMore	0	0	0%	Simple	⌚
Hias	0	0	0%	Simple	⌚
Simple String :D	4	1	25%	Simple	⌚
White Black	2	1	50%	Simple	⌚
Mirana Leap Year	0	0	0%	Simple	⌚

Gambar 4.8 Tampilan List Problem Sets Member

#### h. Tampilan Halaman Solve Problem Member

Tampilan halaman dimana *member* mengirim kode program yang digunakan untuk menyelesaikan soal pemrograman berdasarkan deskripsi. *Member* dapat mengirim kode program dalam bentuk file ataupun mengetik kode program pada *field* yang ada. Tampilan untuk menyelesaikan soal pada kontes ataupun diluar kontes sama. Tampilan dapat dilihat pada Gambar 4.9.



Gambar 4.9 Tampilan Solve Problem Member

i. Tampilan Halaman Create Problem

Tampilan halaman untuk *admin* ataupun *member* membuat soal. Soal yang dibuat oleh *member* akan secara otomatis ter-*publish* dan dapat diselesaikan oleh *member* lain atau dirinya. Dimana soal yang dibuat oleh *admin* tidak ter-*publish* terlebih dahulu kecuali *admin* ingin mem-*publish* dikarenakan *admin* dapat menggunakan soal tersebut untuk kontes. Data pada soal dilengkapi sesuai dengan Gambar 4.10.

**Create a Problem Set**

Name:

Problem ID:

Description :

Category : Dynamic Programming  
Greedy  
Simple

Time Limit (in Seconds):

Memory Limit (in MB):

Testcases Input : Choose File | no file selected

Testcases Output : Choose File | no file selected

**Create Problem**

Gambar 4.10 Tampilan Create Problem

j. Tampilan Halaman Daftar Contest Member

Tampilan halaman daftar dari kontes yang dapat diikuti oleh *member* serta masuk ke dalam kontes yang sedang berjalan. *Member* juga dapat melihat hasil kontes yang telah berjalan sebelumnya. Tampilan daftar kontes untuk *member* dapat dilihat pada Gambar 4.11.

Name	Length	Start Time	End Time	
First Contest	12:00:00	11/6/2016 10:00 AM	11/6/2016 10:00 PM	
Second Contest	8772:00:00	12/6/2016 10:00 AM	12/6/2017 10:00 PM	<b>Go To Contest!</b>

Name	Length	Start Time	End Time	
First Contest	12:00:00	11/6/2016 10:00 AM	11/6/2016 10:00 PM	
Second Contest	8772:00:00	12/6/2016 10:00 AM	12/6/2017 10:00 PM	

Gambar 4.11 Tampilan Daftar Contest Member

### k. Tampilan Halaman Dashboard Admin

Tampilan halaman dimana admin dapat memverifikasi *user*, menambah admin dari *user* ataupun membuat admin baru. Tampilan *dashboard* juga memberikan daftar detil dari *user* yang ada pada sistem. Tampilan dapat dilihat pada Gambar 4.12.

The screenshot shows the Iseloom Admin Dashboard. At the top, there is a navigation bar with links for Iseloom, Dashboard, Universities, Contests, Problem Sets, a search bar labeled 'Search users', and a Logout button. Below the navigation bar, there are two main sections:

- Users**: A table listing five users with columns for No., Name, UserName, Email, University, Date Joined, Activation, and Verification. Each row has a 'Make as Admin' button. The data is as follows:
 

No.	Name	UserName	Email	University	Date Joined	Activation	Verification
1	Denny Ho	moccacino_latte	dennyho58@gmail.com	STMIK Mikroskil	Wed Jun 08 2016 11:42:57 GMT+0700 (WIB)	Activated	Verified
2	Tommy	tommy_cuang	tommy_cuang@yahoo.co.id	STMIK Mikroskil	Wed Jun 08 2016 21:38:09 GMT+0700 (WIB)	Activated	Verified
3	Benny wijaya	Miracle	benny.wijaya8@yahoo.com	STMIK Mikroskil	Thu Jun 09 2016 11:53:35 GMT+0700 (WIB)	Activated	Verified
4	davin	davin	davintardean@gmail.com	STMIK Mikroskil	Thu Jun 09 2016 11:54:06 GMT+0700 (WIB)	Activated	Verified
5	Fredy Ong jaya	Xinvayne	freddy.ong.jaya@gmail.com	STMIK Mikroskil	Thu Jun 09 2016 11:56:42 GMT+0700 (WIB)	Activated	Verified
- Admin**: A table listing two admin users with columns for No., Name, UserName, Email, Date Joined, and Status. Each row has a 'Make as User' button. The data is as follows:
 

No.	Name	UserName	Email	Date Joined	Status
1	Michael Lim	mike	mike@kolexia.com	Thu Jan 28 2016 13:18:18 GMT+0700 (WIB)	Verified
2	Jeffry Tandiono	jt	jeffry.tandiono@mikroskil.ac.id	Sun May 29 2016 12:45:41 GMT+0700 (WIB)	Verified

Gambar 4.12 Tampilan Dashboard Admin

### l. Tampilan Halaman Universitas Admin

Tampilan halaman yang berisi daftar universitas yang digunakan saat *member* melakukan pendaftaran. Admin dapat menambah universitas baru. Tampilan dapat dilihat pada Gambar 4.13.

The screenshot shows the Iseloom Admin Dashboard. At the top, there is a navigation bar with links for Iseloom, Dashboard, Universities, Contests, Problem Sets, a search bar labeled 'Search users', and a Logout button. Below the navigation bar, there are two main sections:

- Universities**: A table listing four universities with columns for No., Name, City, and Country. The data is as follows:
 

No.	Name	City	Country
1	STMIK Mikroskil	Medan	Indonesia
2	Binus University	Jakarta	Indonesia
3	Universitas Indonesia	Jakarta	Indonesia
4	Universitas Prima Indonesia	Medan	Indonesia
- Add University**: A form for adding a new university. It has fields for University Name (Name), City, and Country, each with a dropdown menu. There is also a 'Add University' button.
 

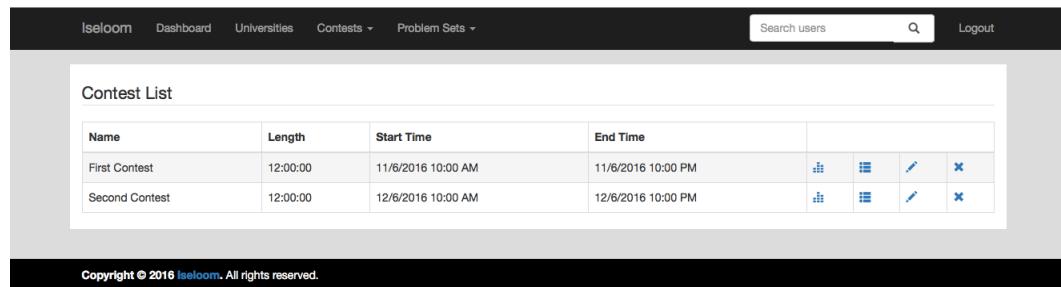
University Name	<input type="text" value="Name"/>
City	<input type="text" value="City"/>
Country	<input type="text" value="Country"/>
<input type="button" value="Add University"/>	

At the bottom of the page, there is a copyright notice: "Copyright © 2016 Iseloom. All rights reserved."

Gambar 4.13 Tampilan Universitas Admin

m. Tampilan Halaman Daftar Contest Admin

Tampilan halaman daftar kontes untuk admin, dimana admin memiliki menu untuk melihat *scoreboard*, mengubah kontes, menghapus kontes serta menambah soal kontes. Tampilan daftar kontes admin dapat dilihat pada Gambar 4.14.

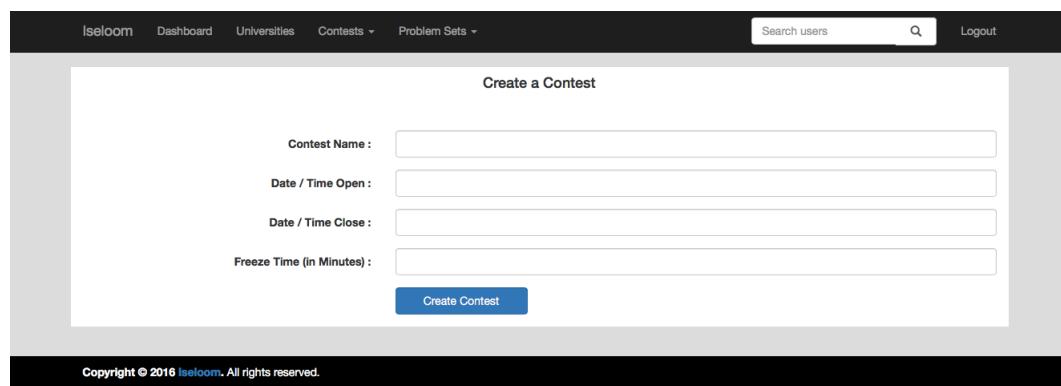


The screenshot shows a web application interface for managing contests. At the top, there is a navigation bar with links for Iseloom, Dashboard, Universities, Contests (with a dropdown arrow), Problem Sets (with a dropdown arrow), a search bar labeled 'Search users' with a magnifying glass icon, and a 'Logout' link. Below the navigation bar is a section titled 'Contest List' containing a table with two rows of contest information. The columns in the table are 'Name', 'Length', 'Start Time', and 'End Time'. Each row has four buttons at the end: a blue square with a white grid, a blue square with a white list, a blue square with a white pencil, and a blue square with a white 'X'. The first row represents 'First Contest' with a start time of 11/6/2016 10:00 AM and an end time of 11/6/2016 10:00 PM. The second row represents 'Second Contest' with a start time of 12/6/2016 10:00 AM and an end time of 12/6/2016 10:00 PM. At the bottom of the page, there is a copyright notice: 'Copyright © 2016 Iseloom. All rights reserved.'

Gambar 4.14 Tampilan Daftar Contest Admin

n. Tampilan Halaman Create Contest

Tampilan halaman dimana admin dapat menambah kontes baru yang akan diadakan. Tampilan halaman *create contest* dapat dilihat pada Gambar 4.15.



The screenshot shows a 'Create a Contest' form. At the top, there is a navigation bar with links for Iseloom, Dashboard, Universities, Contests (with a dropdown arrow), Problem Sets (with a dropdown arrow), a search bar labeled 'Search users' with a magnifying glass icon, and a 'Logout' link. Below the navigation bar is a form titled 'Create a Contest'. It contains four input fields: 'Contest Name' (with a placeholder 'Contest Name'), 'Date / Time Open' (with a placeholder 'Date / Time Open'), 'Date / Time Close' (with a placeholder 'Date / Time Close'), and 'Freeze Time (In Minutes)' (with a placeholder 'Freeze Time (In Minutes)'). At the bottom of the form is a blue 'Create Contest' button. At the very bottom of the page, there is a copyright notice: 'Copyright © 2016 Iseloom. All rights reserved.'

Gambar 4.15 Tampilan Create Contest

o. Tampilan Halaman Problem Contest

Tampilan halaman dimana admin menambah soal pemrograman pada kontes dengan urutan soal. Admin juga dapat melihat kontestan yang sudah mendaftar untuk mengikuti kontes. Tampilan *problem contest* dapat dilihat pada Gambar 4.16.

No	Name	Order
1	Save Mr Oski	1
2	Simple String :D	2
3	White Black	3

No	Name	University
1	Benny wijaya	STMIK Mikroskil
2	Stephanie	STMIK Mikroskil
3	Denny Ho	STMIK Mikroskil

Gambar 4.16 Tampilan Problem Contest

p. Tampilan Scoreboard Contest Running

Tampilan halaman dimana *scoreboard* kontes ditampilkan dengan lengkap baik peserta kontes, soal hingga setiap soal yang telah diselesaikan, dicoba ataupun belum diselesaikan kontestan. Tampilan *scoreboard contest running* dapat dilihat pada Gambar 4.17.

#	Name	Univ	Solved	Score	12	abc	kaip1	hanoi	koinTes
1	jete	Mikroskil	0	0		1	5		
1	mike	Mikroskil	0	0					
1	a	Mikroskil	0	0					

Gambar 4.17 Tampilan Scoreboard Contest Running

q. Tampilan Scoreboard Contest Freeze

Tampilan halaman dimana *scoreboard* kontes telah *freeze* dan perubahan pada *scoreboard* tidak akan ditampilkan hingga kontes

berakhir. Tampilan *scoreboard contest freeze* dapat dilihat pada Gambar 4.18.

The screenshot shows the Iseloom platform interface. At the top, there are navigation links: Iseloom, Submissions, Problem Sets, Contests, and Ranklists. On the right side, there is a search bar labeled "Search users" with a magnifying glass icon and a "Logout" link. The main content area displays a scoreboard titled "Scoreboard MPC Challenge (FREEZE since 10:40:00)". Below the title, it says "00 days 00:16:52". The scoreboard table has columns for #, Name, Univ, Solved, Score, 12, abc, kailpi, hanoi, and koinTes. There are three entries:

#	Name	Univ	Solved	Score	12	abc	kailpi	hanoi	koinTes
1	jete	Mikroskil	1	140		1	6 (40 + 100)		
2	mike	Mikroskil	0	0					
3	a	Mikroskil	0	0					

At the bottom of the page, a copyright notice reads "Copyright © 2016 Iseloom. All rights reserved."

Gambar 4.18 Tampilan Scoreboard Contest Freeze

r. Tampilan Scoreboard Contest Final

Tampilan halaman dimana *scoreboard* kontes ditampilkan dengan lengkap. *Scoreboard* yang ditampilkan tidak *freeze* dan merupakan hasil terakhir dari kontes. Tampilan *scoreboard contest final* dapat dilihat pada Gambar 4.19.

The screenshot shows the Iseloom platform interface. At the top, there are navigation links: Iseloom, Dashboard, Universities, Contests, and Problem Sets. On the right side, there is a search bar labeled "Search users" with a magnifying glass icon and a "Logout" link. The main content area displays a scoreboard titled "Scoreboard First Contest". Below the title, it says "Contest has done." A button labeled "See All Submissions" is visible. The scoreboard table has columns for #, Name, Univ, Solved, Score, pascal\_triangle, artechno\_lamp, guessing\_game, conversion, do\_you\_have\_enough\_money, finite\_loop, and HW. There are eight entries:

#	Name	Univ	Solved	Score	pascal_triangle	artechno_lamp	guessing_game	conversion	do_you_have_enough_money	finite_loop	HW
1	Denny Ho	STMIK Mikroskil	7	819	1 (26)	1 (119)	2 (172 + 20)	2 (79 + 20)	3 (264 + 40)	2 (65 + 20)	1 (34)
2	KX	STMIK Mikroskil	5	1213	1 (214)	2 (232 + 20)		4 (267 + 60)	1 (220)		1 (200)
3	Erick Kwan	STMIK Mikroskil	5	1588	1 (181)	1 (207)		3 (277 + 40)	1 (673)		1 (210)
4	Benny wijaya	STMIK Mikroskil	4	192	3 (10 + 40)	4 (32 + 60)		1 (36)	3		1 (34)
5	Widya Putra	STMIK Mikroskil	3	112	1 (29)	1 (30)	1	1	1	1	1 (33)
6	Tommy	STMIK Mikroskil	3	1103	2 (334 + 20)	1 (363)					7 (346 + 120)
7	Stephanie	STMIK Mikroskil	2	141	3 (62 + 40)		1				2 (39 + 20)
8	Fredy Ong lava	STMIK Mikroskil	2	627	1 (507)	6				3 (80 + 40)	

Gambar 4.19 Tampilan Scoreboard Contest

s. Tampilan Halaman Submissions Contest

Tampilan yang hanya dapat dilihat oleh admin. Admin dapat melihat setiap *submission* yang dilakukan pada kontes dari kontestan, waktu hingga hasil. Hasil tersebut dapat dilihat per *test case* pada *preview*. Admin juga dapat *recompile* kode program user. Tampilan *submissions contest* dapat dilihat pada Gambar 4.20.

The screenshot shows a web application interface for managing contest submissions. At the top, there is a navigation bar with links for 'Iseloom', 'Dashboard', 'Universities', 'Contests', 'Problem Sets', a search bar labeled 'Search users', and a 'Logout' button. Below the navigation bar, the main content area has a title 'All submission for Contest First Contest'. A table lists 10 submissions made by different contestants. The columns are labeled '#', 'ID Problem', 'Contestant', 'Time', 'Result', 'Preview Submission', and 'Recompile'. The 'Result' column uses color coding: green for 'CORRECT', red for 'WRONG ANSWER', and orange for 'SUBMISSION ERROR'. The 'Preview Submission' and 'Recompile' buttons are located in the last two columns of each row.

#	ID Problem	Contestant	Time	Result	Preview Submission	Recompile
1	do_you_have_enough_money	Erick Kwan	Sat Jun 11 2016 21:12:52 GMT+0700 (WIB)	CORRECT	Preview Submission	Recompile
2	artechno_lamp	Fredy Ong jaya	Sat Jun 11 2016 20:12:39 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile
3	artechno_lamp	Fredy Ong jaya	Sat Jun 11 2016 20:09:28 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile
4	artechno_lamp	Fredy Ong jaya	Sat Jun 11 2016 20:00:54 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile
5	artechno_lamp	Fredy Ong jaya	Sat Jun 11 2016 19:15:16 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile
6	artechno_lamp	Fredy Ong jaya	Sat Jun 11 2016 19:03:56 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile
7	artechno_lamp	Fredy Ong jaya	Sat Jun 11 2016 19:02:57 GMT+0700 (WIB)	SUBMISSION ERROR	Preview Submission	Recompile
8	pascal_triangle	Fredy Ong jaya	Sat Jun 11 2016 18:26:42 GMT+0700 (WIB)	CORRECT	Preview Submission	Recompile
9	conversion	Kelvin Hermawan Leonardo	Sat Jun 11 2016 17:08:26 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile
10	conversion	Kelvin Hermawan Leonardo	Sat Jun 11 2016 17:03:46 GMT+0700 (WIB)	WRONG ANSWER	Preview Submission	Recompile

Gambar 4.20 Tampilan Submissions Contest

t. Tampilan Halaman Detail Submission Contest

Tampilan dimana admin melihat secara detil *submission* peserta kontes dari kode program hingga hasil dalam pengujian *test case* kode program. Tampilan halaman detil *submission contest* dapat dilihat pada Gambar 4.21.

Iseloom Dashboard Universities Contests Problem Sets

Search users Logout

## Submission Fredy Ong jaya

**Code**

```
#include <bits/stdc++.h>
int arr[35][35];
int main()
{
    int a, b;
    scanf("%d%d", &a, &b);
    for (int i = 0; i < 30; i++)
    {
        arr[i][0] = 1;
        arr[0][i] = 1;
        for (int j = 1; j < i; j++)
        {
            arr[i][j] = arr[i - 1][j - 1] + arr[i - 1][j];
        }
    }
    printf("%d\n", arr[a - 1][b - 1]);
}
```

**Status**

**CORRECT**

---

**Test Cases**

**Output 1**

```
1
```

**Answer 1**

```
1
```

**Status 1**

**CORRECT**

**Output 2**

```
1
```

**Answer 2**

```
1
```

**Status 2**

**CORRECT**

---

**Output 3**

```
171
```

**Answer 3**

```
171
```

**Status 3**

**CORRECT**

**Output 4**

```
1
```

**Answer 4**

```
1
```

**Status 4**

**CORRECT**

Gambar 4.21 Tampilan Detail Submission Contest

u. Tampilan Halaman Problem Category

Tampilan halaman dimana kategori soal pemrograman ditambah, diubah atau dibuang oleh admin. Kategori digunakan saat membuat soal pemrograman, tampilan dapat dilihat pada Gambar 4.22.

No.	Name	
1	Dynamic Programming	
2	Greedy	
3	Simple	

Gambar 4.22 Tampilan Problem Category

v. Tampilan Halaman Problem Sets Admin

Tampilan halaman dimana admin dapat melihat soal pemrograman yang ada baik yang telah ter-*publish* ataupun yang belum. Soal pemrograman dapat diubah, dihapus ataupun diuji oleh admin. Soal pemrograman yang telah ter-*publish* dapat diselesaikan *member* secara bebas, sedangkan soal yang belum ter-*publish* dapat dengan aman digunakan pada kontes karena belum diketahui oleh *member*. Tampilan *problem sets* admin dapat dilihat pada Gambar 4.23.

NAME	SUBMISSIONS				CATEGORY
	TOTAL	ACC.	RATIO		
Hello World	1	1	100%	Simple	
Show Word	22	8	36%	Simple	
Pascal's Triangle	0	0	0%	Simple	

Not Publish Problem Sets					
NAME					
Artechno Lamp					
Multiplication					

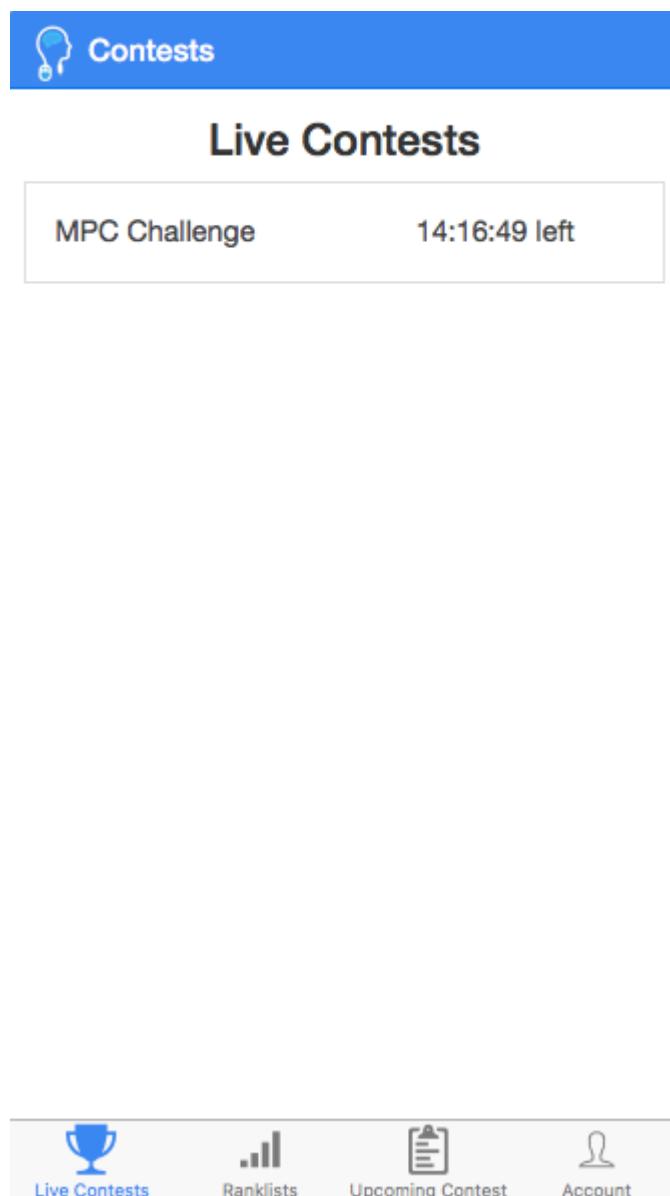
Gambar 4.23 Tampilan Problem Sets Admin

## 2. Tampilan *Mobile*

Tampilan *Mobile* dari *Online Judge* terdiri dari beberapa halaman yaitu sebagai berikut:

### a. Tampilan Mobile Live Contest

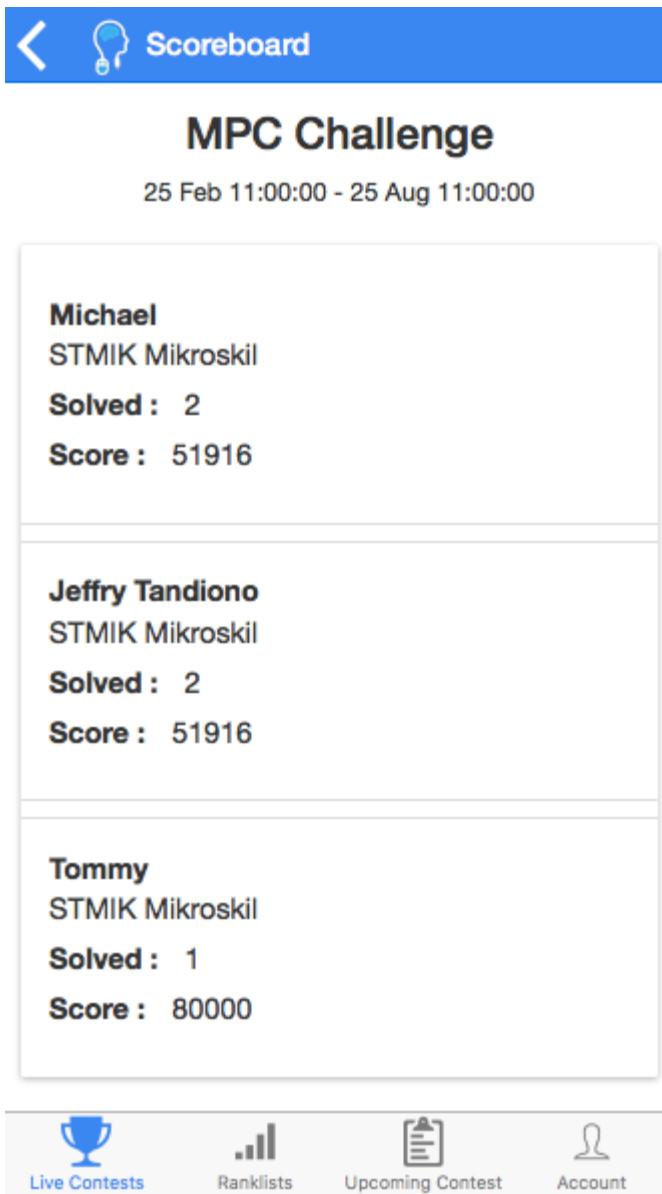
Tampilan dimana pengguna dapat melihat daftar kontes yang sedang berlangsung. Pengguna memilih kontes yang akan dilihat detail *scoreboard* dari kontes tersebut. Tampilan *mobile live contest* dapat dilihat pada Gambar 4.24.



Gambar 4.24 Tampilan Mobile Live Contest

b. Tampilan Mobile Scoreboard

Tampilan dimana pengguna dapat melihat daftar *scoreboard* dari kontes yang dipilih. Pengguna memilih kontestan untuk melihat detil dari soal yang sudah kontestan selesaikan atau belum. Tampilan *mobile scoreboard* dapat dilihat pada Gambar 4.25.



Gambar 4.25 Tampilan Mobile Scoreboard

c. Tampilan Mobile Scoreboard Detail

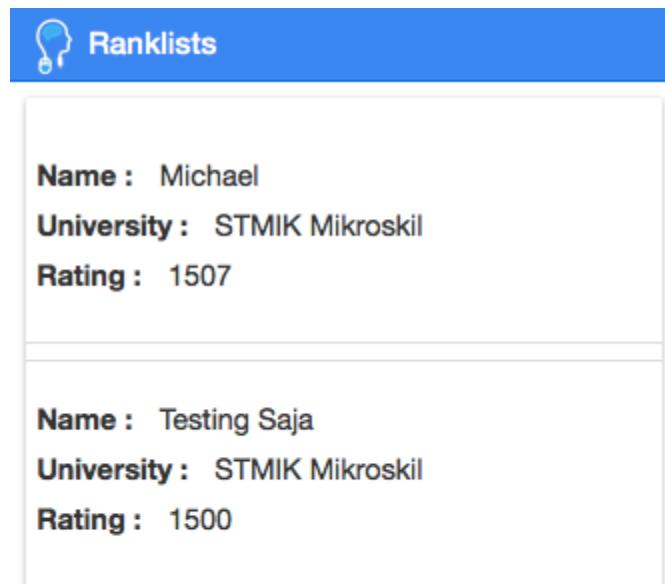
Tampilan dimana pengguna dapat melihat daftar soal yang sudah diselesaikan atau belum dari kontestan yang dipilih. Tampilan mobile scoreboard detail dapat dilihat pada Gambar 4.26.



Gambar 4.26 Tampilan Mobile Scoreboard Detail

d. Tampilan Mobile Ranklist

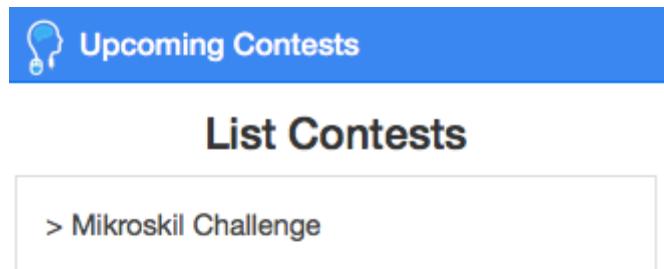
Tampilan dimana pengguna bisa melihat urutan daftar *member* berdasarkan urutan *rating*. Tampilan *mobile ranklist* dapat dilihat pada Gambar 4.27.



Gambar 4.27 Tampilan Mobile Ranklist

e. Tampilan Mobile Upcoming Contest

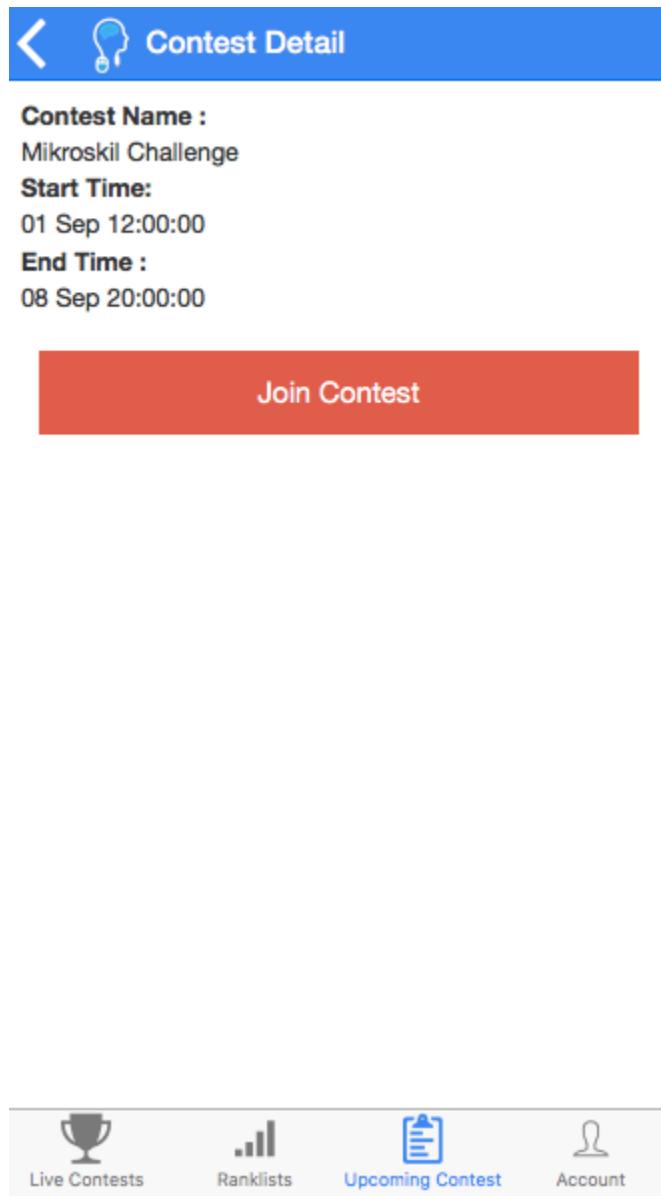
Tampilan dimana pengguna dapat melihat daftar kontes yang akan dimulai. Pengguna memilih kontes untuk melihat detil dari kontes yang akan diikuti atau tidak. Tampilan mobile upcoming contest dapat dilihat pada Gambar 4.28.



Gambar 4.28 Tampilan Mobile Upcoming Contest

f. Tampilan Mobile Upcoming Contest Detail

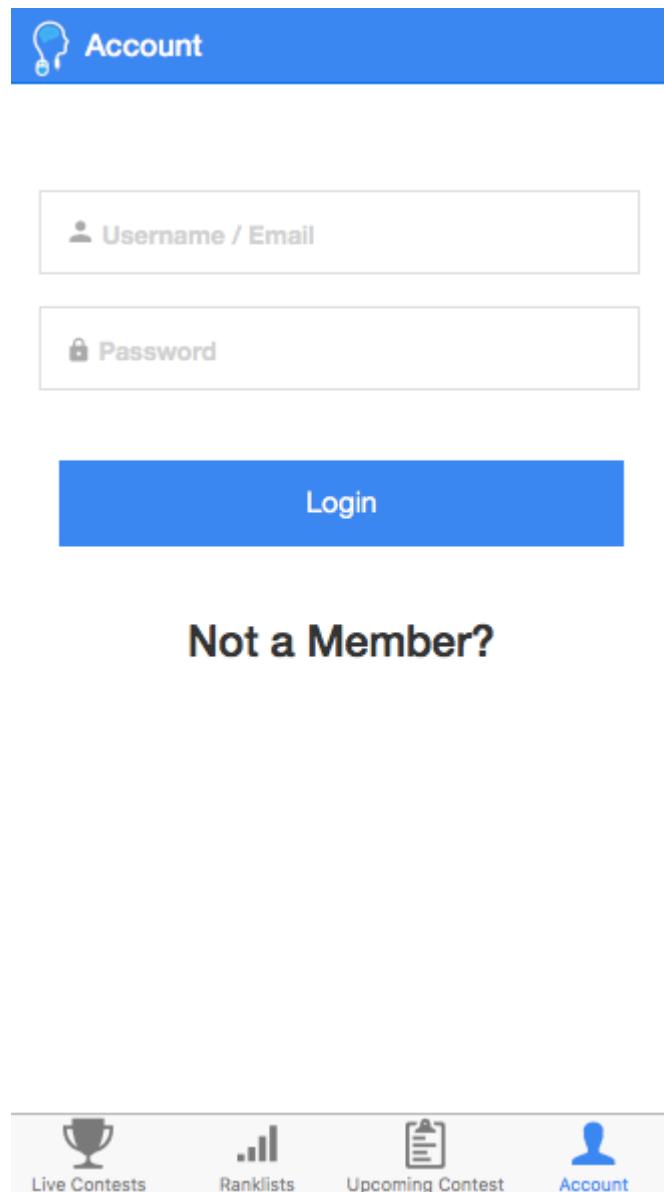
Tampilan dimana pengguna dapat melihat detail dari kontes yang akan dimulai. Pengguna dapat memilih untuk mengikuti kontes atau tidak mengikuti kontes. Tampilan *mobile upcoming contest detail* dapat dilihat pada Gambar 4.29.



Gambar 4.29 Tampilan Mobile Upcoming Contest Detail

g. Tampilan Mobile Login

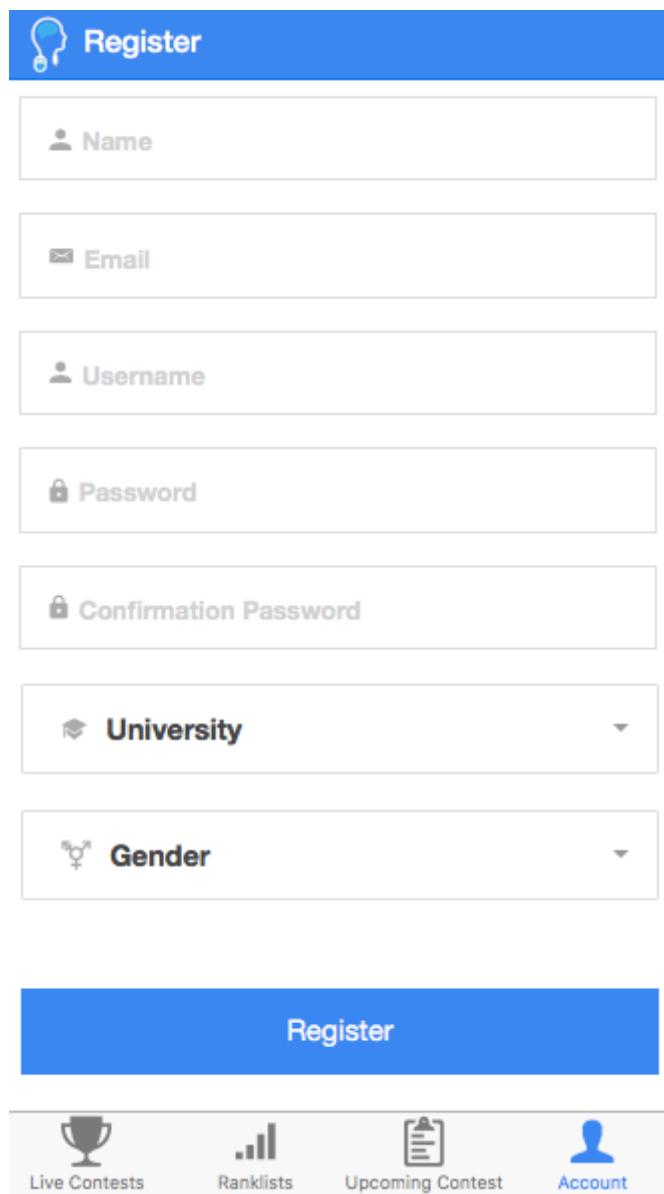
Tampilan dimana pengguna dapat masuk ke dalam sistem apabila memiliki keanggotaan sebagai *member*. Tampilan *mobile login* dapat dilihat pada Gambar 4.30.



Gambar 4.30 Tampilan Mobile Login

h. Tampilan Mobile Register

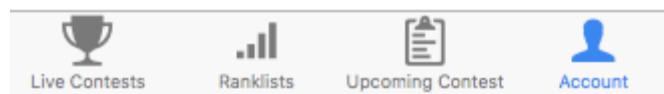
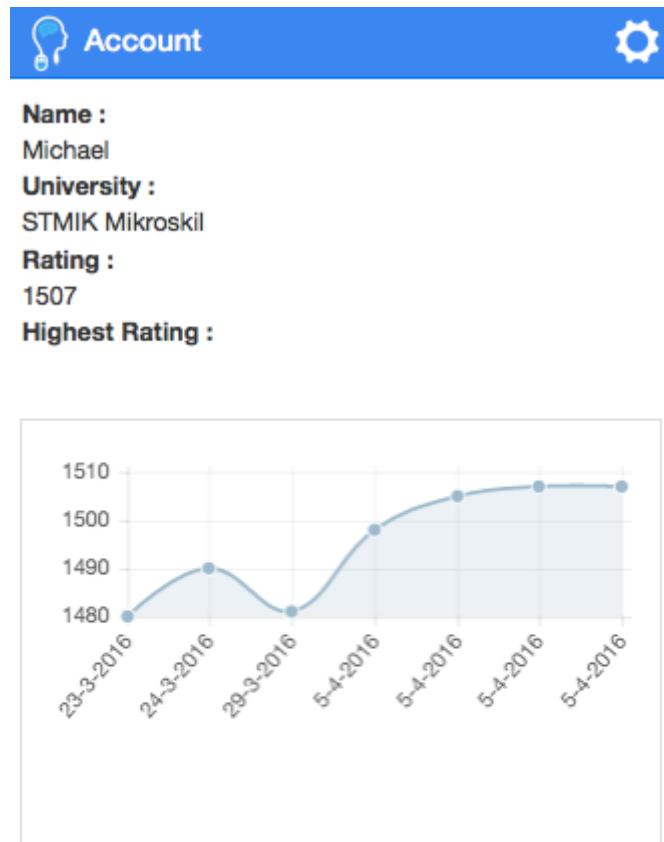
Tampilan dimana pengguna dapat mendaftar diri pada aplikasi untuk dapat menggunakan aplikasi sebagai *member*. Tampilan *mobile register* dapat dilihat pada Gambar 4.31.



Gambar 4.31 Tampilan Mobile Register

i. Tampilan Mobile Account

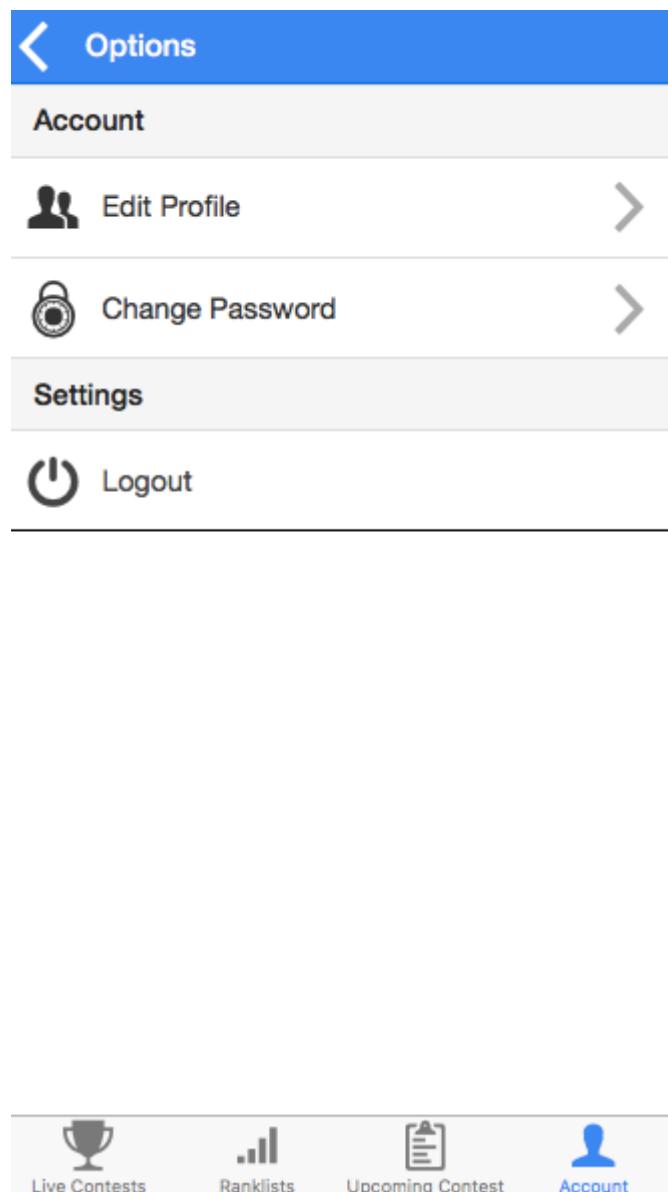
Tampilan dimana pengguna dapat melihat detail dari akun pengguna sendiri. Tampilan *mobile account* dapat dilihat pada Gambar 4.32.



Gambar 4.32 Tampilan Mobile Account

j. Tampilan Mobile Options

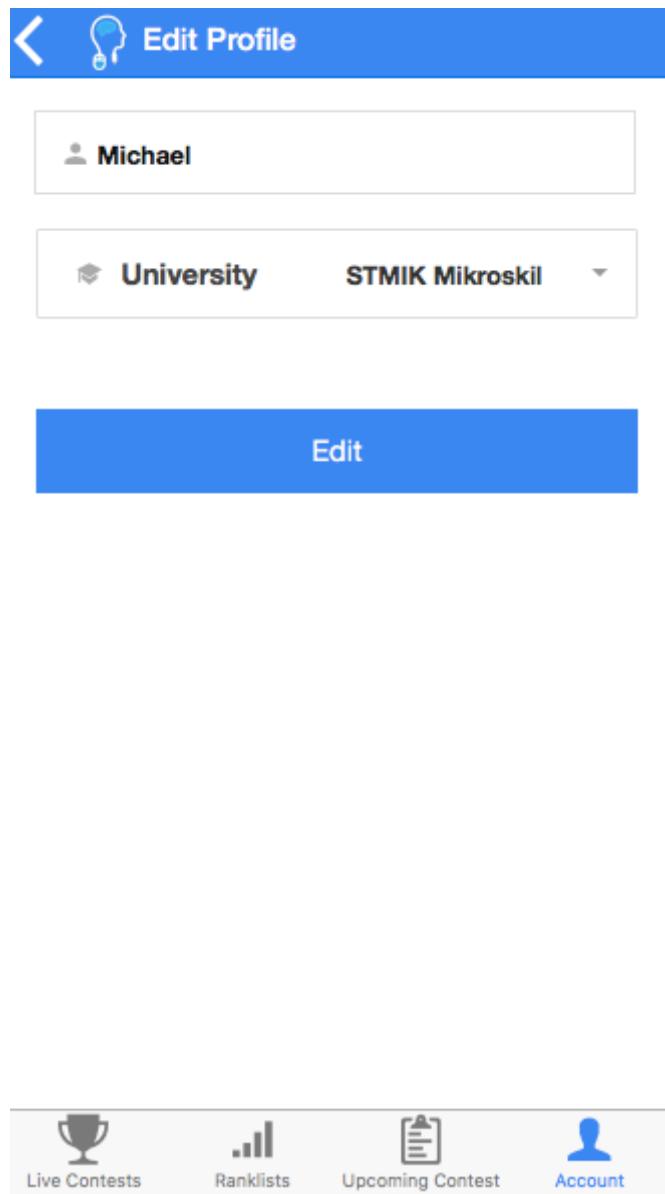
Tampilan dimana pengguna dapat memilih untuk mengedit profil, mengubah *password* atau *logout* dari akun pada aplikasi. Tampilan *mobile options* dapat dilihat pada Gambar 4.33.



Gambar 4.33 Tampilan Mobile Options

k. Tampilan Mobile Edit Profile

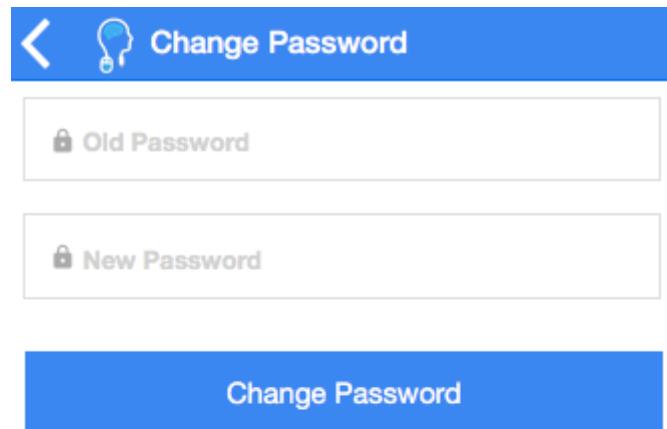
Tampilan dimana pengguna dapat mengedit profil dari akun pengguna sendiri. Pengguna dapat mengubah nama dan universitas dari pengguna sendiri. Tampilan *mobile edit profile* dapat dilihat pada Gambar 4.34.



Gambar 4.34 Tampilan Mobile Edit Profile

1. Tampilan Mobile Change Password

Tampilan dimana pengguna dapat mengubah password dari akun pengguna sendiri. Pengguna memasukkan password lama dan password baru dari pengguna sendiri. Tampilan *mobile change password* dapat dilihat pada Gambar 4.35.



Gambar 4.35 Tampilan Mobile Change Password

## 4.2 Implementasi

Dalam implementasi sistem terhadap pengguna, dilakukan pengujian berupa kontes serta fitur pada sistem. Sistem dapat diakses pada <http://mikelrn.com>. Spesifikasi dari *server* sistem dapat dilihat pada Tabel 4.1.

Tabel 4.1 Spesifikasi Server

<b>Server</b>	Ubuntu 14.04
<b>Memori</b>	1024 MB
<b>Kapasitas Data</b>	30 GB
<b>Penyedia</b>	Digital Ocean
<b>Lokasi</b>	Singapore

Pengujian tingkat kepuasan dan penggunaan sistem dilakukan dengan mengadakan kontes sebanyak 2 kali dan membuka fitur sistem yang dapat digunakan penguji sistem kapanpun. Daftar kontes dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar Kontes

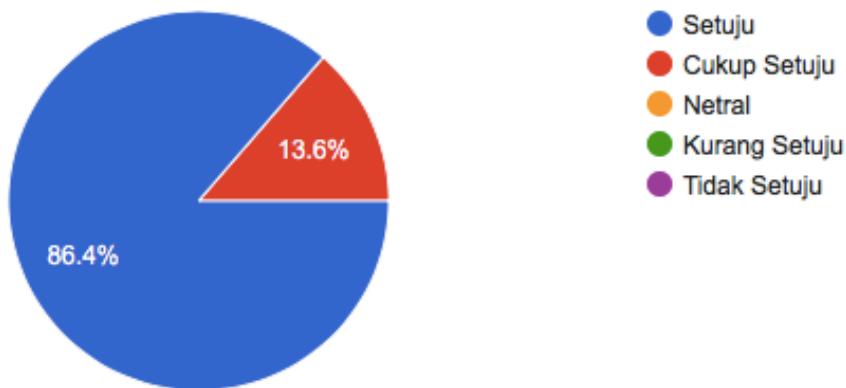
Nama	Hari, Tanggal	Jam	Peserta
First Contest	Sabtu, 11 Juni 2016	10.00 - 22.00	23
Second Contest	Minggu, 11 Juni 2016	10.00 - 22.00	24

Pengujian tingkat kepuasan dan penggunaan sistem dilakukan dengan menggunakan kuesioner. Kuesioner dilakukan dengan Google Forms yang dapat diakses pada <http://goo.gl/forms/seTHtqJC9VyBbZQ23>. Tampilan kuesioner dapat dilihat pada Lampiran 1.

Dari kuesioner tersebut, hasil yang diterima didapatkan dari 22 responden. Hasil kuesioner dibagi per pertanyaan dengan 5 kriteria jawaban yang sama, antara lain :

1. Pendaftaran aplikasi cukup mudah dan membantu.

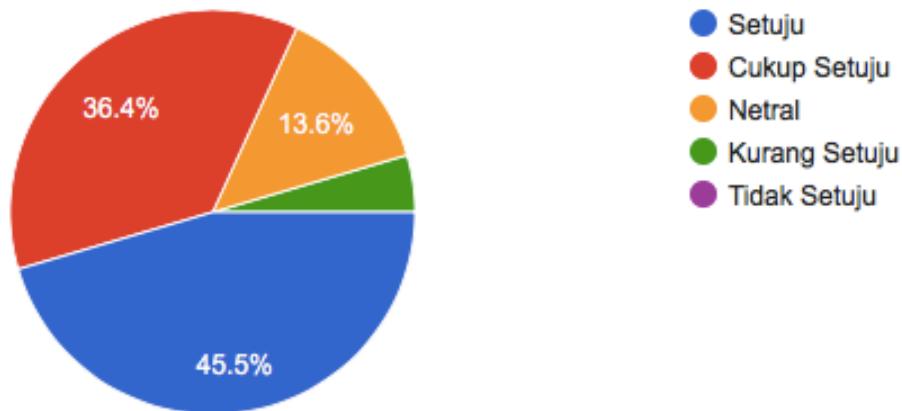
Pernyataan ini membahas pendaftaran yang dilakukan oleh pengguna. Semakin mudah pendaftaran dilakukan akan semakin menarik minat pengguna yang menggunakan. Hasil dapat dilihat pada Gambar 4.36.



Gambar 4.36 Hasil Kuesioner Pernyataan 1

2. Tampilan aplikasi bagus dan menarik minat pengguna.

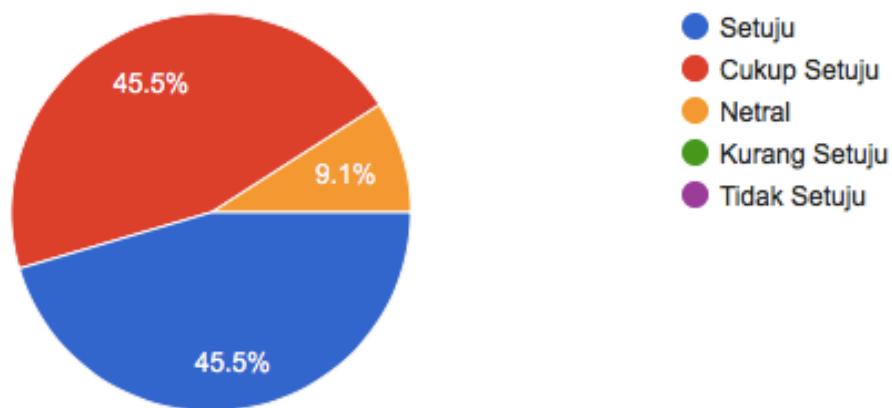
Pernyataan ini membahas tampilan aplikasi secara keseluruhan. Tampilan cukup subjektif digunakan sebagai kuesioner tetapi nilai jual sebuah aplikasi dari tampilan cukup besar. Hasil dapat dilihat pada Gambar 4.37.



Gambar 4.37 Hasil Kuesioner Pernyataan 2

3. Typografi aplikasi bagus dan menarik.

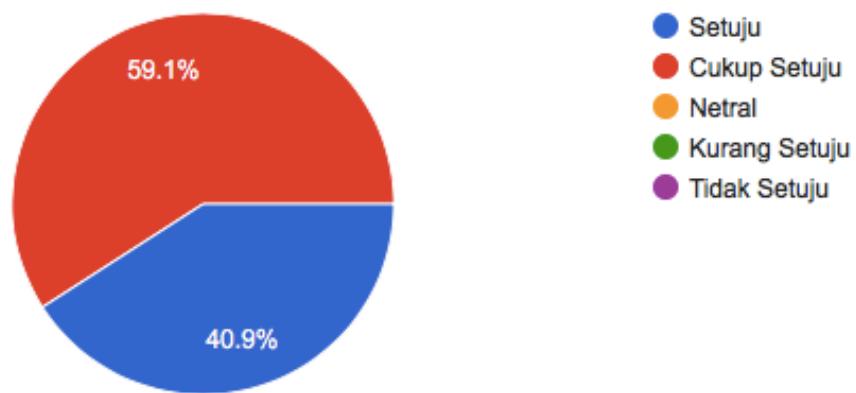
Pernyataan ini membahas typografi dari aplikasi, dari ukuran hingga ketebalan tulisan. Pada kontes pemrograman tulisan yang mudah dibaca sangat berguna membantu, dikarenakan kebutuhan tersebut dilakukanlah kuesioner dengan pernyataan ini. Hasil dapat dilihat pada Gambar 4.38.



Gambar 4.38 Hasil Kuesioner Pernyataan 3

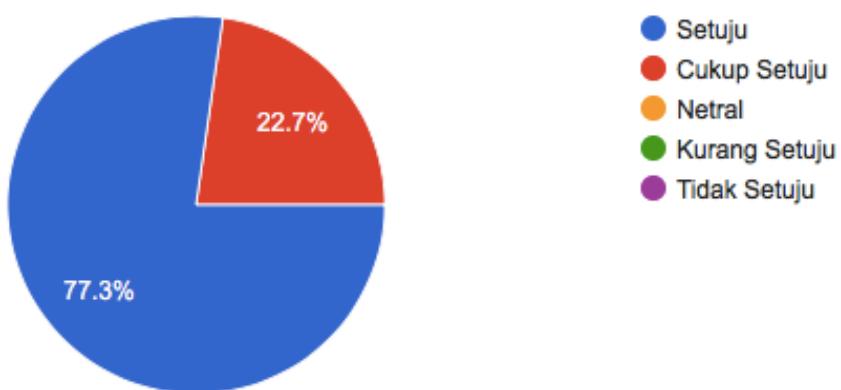
4. Menu pada aplikasi cukup dan membantu.

Menu pada sebuah aplikasi merupakan jalan pintas yang dibuat untuk membantu pengguna dalam menggunakan aplikasi. Menu tidak saja membantu pengguna berinteraksi dengan aplikasi, sekaligus memberitahukan pengguna akan fitur yang ada. Hasil dapat dilihat pada Gambar 4.39.



Gambar 4.39 Hasil Kuesioner Pernyataan 4

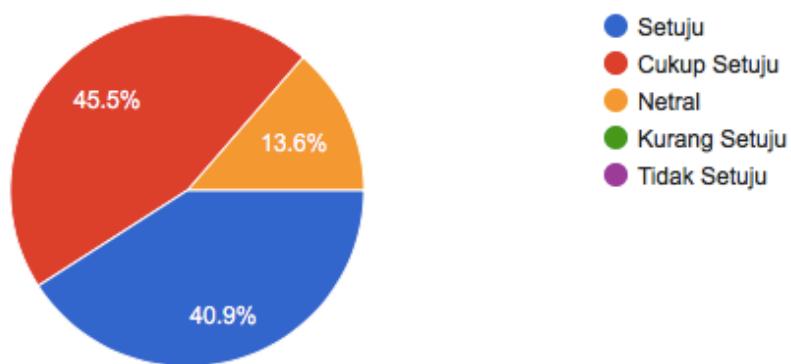
5. Fitur pengguna dapat membuat soal mampu membantu menambah bank soal sekaligus membantu user berlatih dalam menghasilkan soal yang baik. Pernyataan ini menjelaskan fitur tambahan dari aplikasi *online judge* yaitu memperbolehkan pengguna membuat soal. Fitur ini mampu membantu bank soal pada aplikasi terus bertambah sekaligus menjadi wadah pengguna berlatih dalam membuat soal. Hasil dapat dilihat pada Gambar 4.40.



Gambar 4.40 Hasil Kuesioner Pernyataan 5

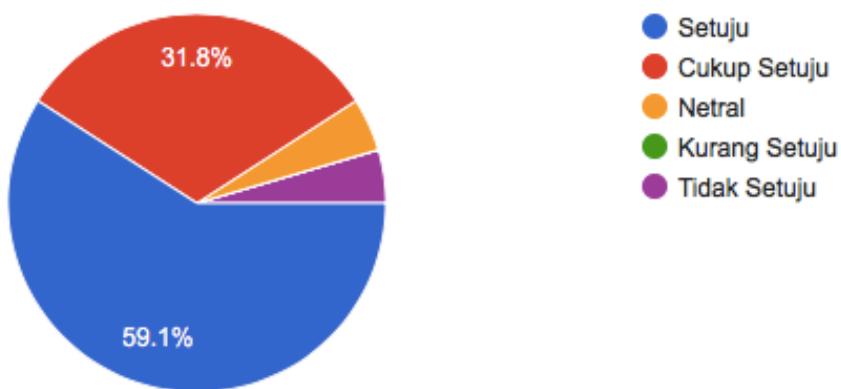
6. Keamanan aplikasi terjamin dan terpercaya.

Pernyataan ini membahas tentang keamanan dari aplikasi baik. Keamanan merupakan faktor penting dari aplikasi, semakin tidak amannya sebuah aplikasi membuat pengguna menghindari penggunaan aplikasi. Hasil dapat dilihat pada Gambar 4.41.



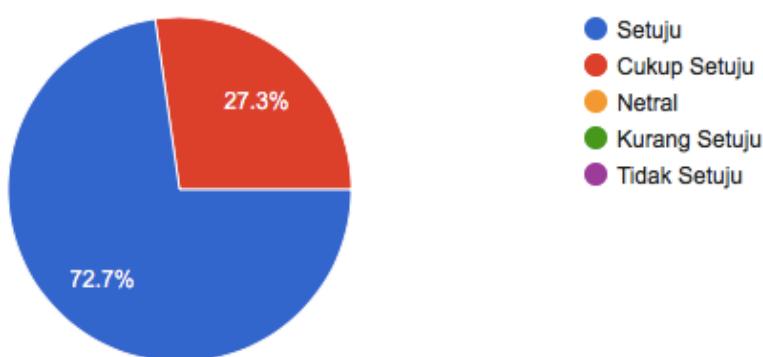
Gambar 4.41 Hasil Kuesioner Pernyataan 6

7. Rating anda bila dibandingkan dengan kontestan lain, adil dan terpercaya. Pernyataan ini sepenuhnya membahas tentang *rating* yang didapatkan oleh pengguna. *Rating* akan berubah bila pengguna mengikuti kontes, *rating* berubah sesuai dengan hasil yang didapatkan oleh pengguna pada kontes. Hasil dapat dilihat pada Gambar 4.42.



Gambar 4.42 Hasil Kuesioner Pernyataan 7

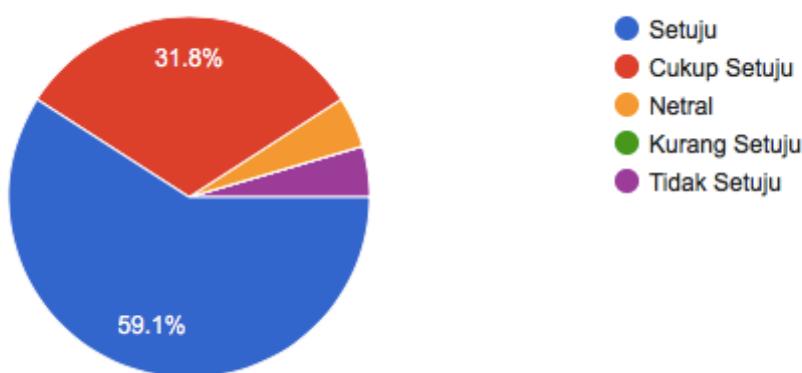
8. History rating pada profil anda dapat membantu tracking rating anda. Pernyataan ini membahas fitur untuk *tracking* pelajar dari *history rating* milik pengguna. Perubahan dari *rating* dapat menjadi tolak ukur kemampuan pengguna dari waktu ke waktu. Hasil dapat dilihat pada Gambar 4.43.



Gambar 4.43 Hasil Kuesioner Pernyataan 8

9. Penjurian cukup akurat dan cepat.

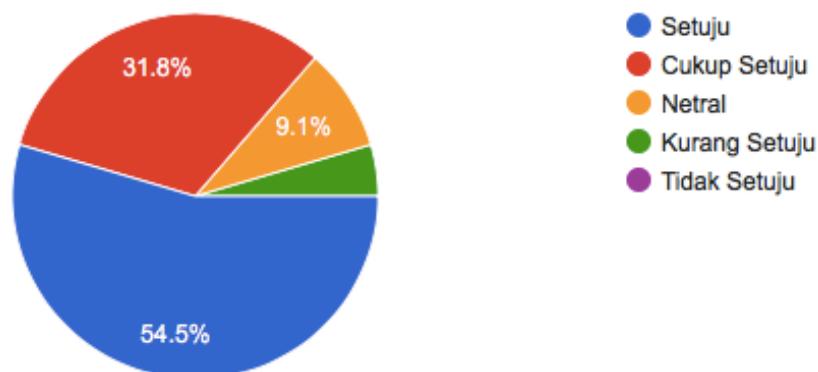
Pernyataan ini membahas tentang penjurian yang dilakukan oleh *compiler* terhadap kode program yang dikirimkan oleh pengguna. Dalam menghasilkan *online judge* yang baik diperlukan penjurian yang akurat dan cepat. Hasil dapat dilihat pada Gambar 4.44.



Gambar 4.44 Hasil Kuesioner Pernyataan 9

10. Aplikasi cukup nyaman dan gampang digunakan.

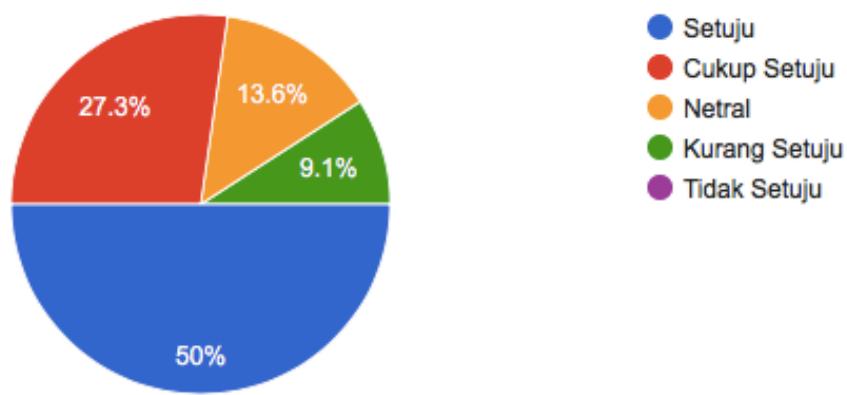
Pernyataan ini membahas aplikasi secara keseluruhan. Aplikasi yang nyaman dan gampang digunakan lebih disukai dan sering digunakan pengguna dibandingkan aplikasi yang sulit digunakan. Hasil dapat dilihat pada Gambar 4.45.



Gambar 4.45 Hasil Kuesioner Pernyataan 10

11. Aplikasi secara keseluruhan cukup baik untuk menjadi online judge yang digunakan pada kontes pemrograman seperti ideafuse.

Pernyataan ini membandingkan aplikasi yang dihasilkan dengan kontes pemrograman untuk menjadi tolak ukur kemampuan aplikasi yang dihasilkan. Sebagai aplikasi *online judge* yang dapat dipercaya dan bagus dalam digunakan pada kontes pemrograman seperti ideafuse memiliki banyak faktor. Hasil dapat dilihat pada Gambar 4.46.



Gambar 4.46 Hasil Kuesioner Pernyataan 11

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian, beberapa kesimpulan penting yang dapat dikemukakan tentang aplikasi *online judge* dan *tracking* pelajar adalah sebagai berikut:

1. Dalam *tracking* pelajar, perhitungan Elo Rating dapat dipercaya dengan 59.1% Setuju dan 31.8% Cukup Setuju dari hasil kuesioner. Dengan menggunakan kontes pemrograman, pengaruh *rating* peserta akan menghasilkan *history* yang dapat digunakan pada *tracking rating* mendapatkan 72.7% Setuju dan 27.3% Cukup Setuju dari hasil kuesioner.
2. Terdapat 77,3% pengguna yang setuju dan 22,7% pengguna cukup setuju untuk pernyataan fitur membuat soal mampu menambah bank soal pada *online judge* dari hasil kuesioner.

#### **5.2 Saran**

Adapun saran untuk pengembangan selanjutnya berkaitan dengan aplikasi *online judge* dan *tracking* pelajar yang dirancang yaitu :

1. Aplikasi perlu ditambahkan beberapa Bahasa seperti Java, Python dan Ruby untuk meningkatkan pengguna *online judge* yang menggunakan bahasa selain C dan C++.
2. Tampilan dan fungsionalitas aplikasi baiknya mengikuti aturan ACM ICPC. Pengguna mampu mengadaptasi apabila aturan serupa dengan aturan ACM ICPC yang telah banyak diadaptasi oleh kontes pemrograman di dunia.

## **DAFTAR PUSTAKA**

- Caterineum, A, Petit, J. 2011. Jutge.org: Instructor's Guide.  
<http://www.jutge.org/documentation/guide.pdf>.
- Glickman, M. 2001. Approximating Formulas for the USCF Rating System.  
Boston.
- Halim, S, Halim, F. 2013. Competitive Programming, 3rd Edition.
- Hallinan, M. 1994. Tracking: From Theory to Practice.
- Hartmann, AK. 2009. Practical Guide to Computer Simulations. Singapore.
- Ilyumzhinov, K. 2008. FIDE Rating Regulations. United States.
- Kerninghan, B. 1999. The Practice of Programming. Pearson.
- Leal, JP, Silva, F. 2003. Mooshak: a Web-based multi-site programming contest system. Portugal.
- Manzoor, S. 2006. Analyzing Programming Contest Statistics. CLIS, San Antonio.
- Revilla, MA, Manzoor, S, Liu, R. Competitive Learning in Informatics: The UVa Online Judge Experience. Spain. 2008.
- Silver, A, Anthon, C. 2007. All About GNU Backgammon.
- Silver, N, Fischer-Baum, R. 2015. How We Calculate NBA Elo Ratings.
- Skeina, S, Revilla, MA. 2003. Programming Challenges: The Programming Contest Training Manual. New York : Springer. 2003.  
<http://codeforces.com/blog/entry/20762>.

# LAMPIRAN 1

## KUESIONER

### Kuesioner Iseloom

Pilih pilihan yang paling sesuai..

\* Required

**Tuliskan nama anda yang digunakan pada pengujian Iseloom \***

Your answer \_\_\_\_\_

**Pendaftaran aplikasi cukup mudah dan membantu \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Tampilan aplikasi bagus dan menarik minat pengguna \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Typografi aplikasi bagus dan menarik \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Menu pada aplikasi cukup dan membantu \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Fitur pengguna dapat membuat soal mampu membantu menambah bank soal sekaligus membantu user berlatih dalam menghasilkan soal yang baik \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Keamanan aplikasi terjamin dan terpercaya \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Rating anda bila dibandingkan dengan kontestan lain, adil dan terpercaya \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**History rating pada profil anda dapat membantu tracking rating anda \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Penjurian cukup akurat dan cepat \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Aplikasi cukup nyaman dan gampang digunakan \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Aplikasi secara keseluruhan cukup baik untuk menjadi online judge yang digunakan pada kontes pemrograman seperti ideafuse \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

#### **Saran**

Your answer

---

#### **Komentar**

Your answer

---

**SUBMIT**

Never submit passwords through Google Forms.

## LAMPIRAN 2

### SAMPEL KUESIONER 1

#### Kuesioner Iseloom

Pilih pilihan yang paling sesuai..

\* Required

Tuliskan nama anda yang digunakan pada pengujian Iseloom \*

Xinvayne (Fredy Ong Jaya)

Pendaftaran aplikasi cukup mudah dan membantu \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Tampilan aplikasi bagus dan menarik minat pengguna \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Typografi aplikasi bagus dan menarik \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Menu pada aplikasi cukup dan membantu \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Fitur pengguna dapat membuat soal mampu membantu menambah bank soal sekaligus membantu user berlatih dalam menghasilkan soal yang baik \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Keamanan aplikasi terjamin dan terpercaya \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Rating anda bila dibandingkan dengan kontestan lain, adil dan terpercaya \***

- Setuju  
 Cukup Setuju  
 Netral  
 Kurang Setuju  
 Tidak Setuju

**History rating pada profil anda dapat membantu tracking rating anda \***

- Setuju  
 Cukup Setuju  
 Netral  
 Kurang Setuju  
 Tidak Setuju

**Penjurian cukup akurat dan cepat \***

- Setuju  
 Cukup Setuju  
 Netral  
 Kurang Setuju  
 Tidak Setuju

**Aplikasi cukup nyaman dan gampang digunakan \***

- Setuju  
 Cukup Setuju  
 Netral  
 Kurang Setuju  
 Tidak Setuju

**Aplikasi secara keseluruhan cukup baik untuk menjadi online judge yang digunakan pada kontes pemrograman seperti ideafuse \***

- Setuju  
 Cukup Setuju  
 Netral  
 Kurang Setuju  
 Tidak Setuju

#### **Saran**

1. bisa lihat hasil submission(timelimit dll)
2. pada submission ada link ke soal yg di submit pada bagian kiri
3. dashboard kontes(alternatif) atau header daftar soal pada masing2 soal kontes
4. tambah pesan compiler error dan memory limit exceeded
5. cek css login untuk tampilan 1366x768.. wkwkk
6. tambah faq dan info compiler, server, dll :p
7. lupa apa lagi
8. sekian dan terimakasih

#### **Komentar**

ggwp

## LAMPIRAN 3

### SAMPEL KUESIONER 2

#### Kuesioner Iseloom

Pilih pilihan yang paling sesuai..

\* Required

Tuliskan nama anda yang digunakan pada pengujian Iseloom \*

Erick Kwan .....

Pendaftaran aplikasi cukup mudah dan membantu \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Tampilan aplikasi bagus dan menarik minat pengguna \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Typografi aplikasi bagus dan menarik \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Menu pada aplikasi cukup dan membantu \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Fitur pengguna dapat membuat soal mampu membantu menambah bank soal sekaligus membantu user berlatih dalam menghasilkan soal yang baik \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

Keamanan aplikasi terjamin dan terpercaya \*

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Rating anda bila dibandingkan dengan kontestan lain, adil dan terpercaya \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**History rating pada profil anda dapat membantu tracking rating anda \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Penjurian cukup akurat dan cepat \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Aplikasi cukup nyaman dan gampang digunakan \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

**Aplikasi secara keseluruhan cukup baik untuk menjadi online judge yang digunakan pada kontes pemrograman seperti ideafuse \***

- Setuju
- Cukup Setuju
- Netral
- Kurang Setuju
- Tidak Setuju

#### **Saran**

Tampilan pada ScoreBoard terlalu panjang, sehingga membuat tampilan sedikit kacau saat resolusi kecil.

Saat loading pada ScoreBoard dan submission, terlihat code-code aneh. Disarankan kedepannya code-code tersebut tidak keluar lagi, agar terlihat lebih mulus.

Title Page pada setiap soal selalu sama, disarankan menggunakan title sesuai judul Problem soal.

---

#### **Komentar**

Menurutku, untuk sekarang rating masih 7/10, jika tampilan dan error-error yang lain telah diperbaiki, mungkin bisa sampai 9/10. Aku masih penasaran pada bagian submission. Saat submit, selalu compiling, bahkan sampai kontes berakhir. Semoga hasil ini tidak membuahkan hasil yang buruk, Good Luck & Happy Coding~~

## **DAFTAR RIWAYAT HIDUP**

**Nama** : Jeffry Tandiono  
**Umur** : 22 tahun  
**Tempat / Tanggal Lahir** : Medan / 22 Mei 1994  
**Jenis Kelamin** : Pria  
**Agama** : Buddha  
**Tempat tinggal** : Jln. Prof H.M. Yamin S.H. No. 200C, Medan

**Pendidikan :**

Tamat SD Perguruan Methodist-3 Medan	Tahun 2008
Tamat SMP Perguruan Methodist-3 Medan	Tahun 2010
Tamat SMA Perguruan Methodist-3 Medan	Tahun 2012
STMIK-STIE MIKROSKIL, Jurusan Teknik Informatika - Sedang dijalani.	

Demikian daftar riwayat hidup ini saya perbuat dengan sesungguhnya.

Hormat saya,



Jeffry Tandiono

## **DAFTAR RIWAYAT HIDUP**

Nama	:	Michael
Umur	:	22 tahun
Tempat/Tanggal Lahir	:	Medan / 03 Maret 1994
Jenis Kelamin	:	Pria
Agama	:	Buddha
Tempat tinggal	:	Jl.Sampali No.33A, Medan

**Pendidikan :**

Tamatian SD Perguruan Methodist-3 Medan	Tahun 2008
Tamatian SMP Perguruan Methodist-3 Medan	Tahun 2010
Tamatian SMA Perguruan Methodist-3 Medan	Tahun 2012
STMIK-STIE MIKROSKIL, Jurusan Teknik Informatika - Sedang dijalani	

Demikian daftar riwayat hidup ini saya perbuat dengan sesungguhnya.

Hormat saya,



Michael