



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Beberapa metode penelitian yang digunakan untuk melakukan penelitian ini antara lain.

1. Studi Literatur

Dalam studi literatur, dilakukan pembelajaran terhadap teori-teori yang berhubungan dengan pengimplementasian gamifikasi. Teori-teori tersebut antara lain adalah pemrograman kompetitif, *online judge*, gamifikasi, dan elemen gamifikasi.

2. Perancangan Sistem

Dalam perancangan sistem, dilakukan beberapa perancangan antara lain perancangan elemen gamifikasi yang akan diimplementasi, perancangan *flowchart*, *Entity Relation Diagram* (ERD), *database schema*, struktur tabel, dan *user interface* dari sistem yang akan dibuat.

3. Implementasi

Dalam implementasi, elemen gamifikasi, *flowchart*, ERD, *database schema*, struktur tabel dan *user interface* yang telah dirancang sebelumnya akan diterapkan pada UMN OJ.

4. Pengujian

Dalam pengujian, UMN OJ yang telah diimplementasikan gamifikasi akan dilakukan uji coba dengan meminta anggota UMNPc untuk mencoba menggunakan UMN OJ tersebut dalam latihan pemrograman kompetitif.

5. Evaluasi

Dalam evaluasi, akan disebar kuesioner yang mengacu pada model *Hedonic Motivation System Adoption Model* (HMSAM) untuk mengukur tingkat penerimaan *user* dan mengambil data *submission* yang masuk selama masa pengujian untuk mengukur tingkat partisipasi.

3.2 Analisis Gamifikasi

Berdasarkan *framework* desain gamifikasi yang diajukan Werbach dan Hunter (2012), enam tahapan dalam merancang gamifikasi pada UMN Online Judge yaitu, *define business objectives, delineate target behaviors, describe the players, devise activity loops, don't forget the fun, dan deploy the appropriate tools.*

a) *Define business objectives*

Tujuan utama dari UMN OJ yaitu dapat menguji kemampuan semua anggota UMNPC yang telah mempelajari materi pemrograman kompetitif serta dapat meningkatkan kemampuan anggotanya. Menurut hasil wawancara dengan ketua UMNPC, jumlah *user* yang mengerjakan soal dan rata-rata soal yang dikerjakan setiap *user* masih kurang (Liviani, 2017), menunjukkan tingkat partisipasi yang masih kurang. Kurangnya tingkat partisipasi membuat UMN OJ tidak dapat menguji kemampuan semua anggota UMNPC karena tidak semua anggota menggunakan UMN OJ dalam berlatih pemrograman kompetitif. Tujuan implementasi gamifikasi pada UMN OJ adalah untuk meningkatkan tingkat partisipasi anggota UMNPC dalam mengerjakan soal-soal yang ada di UMN OJ. Dengan meningkatnya partisipasi diharapkan kemampuan *problem solving* anggota juga meningkat sehingga UKM UMNPC dapat meraih prestasi lebih baik lagi dalam mengikuti kompetisi pemrograman kompetitif.

b) *Delineate target behaviors*

Perilaku-perilaku yang menjadi target untuk dilakukan oleh *user* serta cara mengukurnya akan dijelaskan sebagai berikut.

- Melakukan registrasi ke sistem. Dapat diukur dengan melihat data di *database*. Untuk dapat mengerjakan soal yang ada di UMN OJ, *user* perlu mendaftar terlebih dahulu karena jika belum terdaftar, maka soal-soal hanya bisa dilihat dan jawaban dari soal tersebut tidak dapat di-*submit*.
- Mengerjakan semua soal yang ada di UMN OJ. Dapat diukur dengan melihat pada *submission* yang masuk ke UMN OJ. Dengan aktif mengerjakan semua soal yang ada, partisipasi anggota di UMN OJ dapat meningkat.
- Menemukan cara-cara baru dalam menyelesaikan soal-soal yang ada. Dapat diukur dengan melihat pada *submission* yang masuk ke UMN OJ. Partisipasi anggota di UMN OJ dapat tetap meningkat ketika *user* yang telah mengerjakan semua soal yang ada, *user* mengerjakan lagi soal-soal tersebut dengan cara yang baru. Sehingga selain partisipasi anggota tetap meningkat, kemampuan *problem solving* anggota juga meningkat.
- Mengumpulkan sesuatu yang menjadi tanda pencapaian selama berada di UMN OJ. Dapat diukur dengan melihat pada profil pengguna. Untuk mengumpulkan semua tanda-tanda pencapaian yang ada di sistem, *user* harus aktif menggunakan sistem, terutama dalam mengerjakan soal.
- Aktif berpartisipasi pada *comment section* di soal-soal yang ada. Dapat diukur dengan melihat pada *comment section* yang ada di setiap soal. Ketika *user* aktif berinteraksi di *comment section*, berarti *user* juga aktif menggunakan sistem.

- Aktif mengundang orang lain untuk bergabung di UMN OJ. Dapat diukur dengan melihat data di *database*. Dengan mengundang orang lain bergabung di UMN OJ, maka semakin banyak *user* yang dapat berpartisipasi mengerjakan soal-soal di UMN OJ.

c) *Describe the players*

Orang-orang yang akan berpartisipasi yaitu anggota UMNPC. Berdasarkan hasil Bartle Test yang telah dilakukan pada lampiran nomor 5, diperoleh tipe pemain anggota UMNPC didominasi oleh tipe *Explorer* dan *Socializer*. Elemen *game* yang diprediksi untuk populasi tersebut yaitu, elemen *gifting* untuk tipe pemain *socializer* berupa *hint point* yang dapat diperoleh dari *like* pada *comment section* dan *invite user*, dimana *hint point* tersebut dapat digunakan untuk membuka *hint* yang ada pada *problem*. Elemen *resource acquisition*, *hidden badge*, dan *content unlocking* untuk tipe pemain *explorer*.

d) *Devise activity loops*

Cara memotivasi pemain yang dibagi dalam *engagement* dan *progression loop* adalah sebagai berikut.

- *Engagement loop*

Motivation: terdapat soal-soal yang dapat dikerjakan untuk mengetahui sejauh mana *user* telah mengerti suatu materi.

Action: *user* mengerjakan soal yang ada, yang menghasilkan *feedback*.

Feedback: *user* mendapatkan *feedback* berupa XP apabila berhasil mengerjakan soal yang ada, *badges* apabila berhasil mencapai *achievement*, atau telah mencapai XP sejumlah tertentu dan memenuhi XP untuk naik *level*.

- *Progression loop*

Terdapat tingkatan *progress* dari *user* selama berada dalam UMN OJ yang direpresentasikan dengan *level*, dimana materi pembelajaran di tiap *level* disesuaikan dengan kurikulum yang ada di UMN OJ seperti yang dapat dilihat pada lampiran nomor 6. XP yang dibutuhkan di masing-masing *level* cenderung akan meningkat. Namun, pada beberapa *level* tertentu XP yang dibutuhkan untuk naik ke *level* berikutnya lebih kecil atau diturunkan. Hal ini dilakukan karena tingkat kesulitan dari materi yang dipelajari di tiap *level* juga terus meningkat. Sehingga pada saat memasuki *level* dengan tingkat kesulitan materi meningkat, maka XP yang dibutuhkan untuk naik ke *level* selanjutnya akan lebih sedikit. Hal ini dilakukan agar *user* tidak merasa jemu dan memungkinkan *user* akan merasa senang melihat pencapaian yang telah diperoleh selama berada di UMN OJ.

- e) *Don't forget the fun*

Aspek *game* yang tetap memotivasi *user* adalah sebagai berikut.

- *Resource acquisition*, karena seorang *user* dengan tipe pemain *explorer* ingin menemukan *item* yang bisa dikoleksi dan berguna.
- *Content unlocking*, karena seorang *user* dengan tipe pemain *explorer* ingin menemukan soal baru yang diperoleh dengan tidak sengaja.
- *Hidden badge*, karena seorang *user* dengan tipe pemain *explorer* ingin mendapatkan *badges* yang tidak diketahui sebelumnya.
- *Comment section*, karena seorang *user* dengan tipe pemain *socializer* merasa senang dapat berinteraksi dengan *user* lain. Apakah memberi pertanyaan maupun memberikan jawaban yang dapat membantu *user* lain.

- *Gifting* berupa *Hint point* yang dapat diperoleh dari *Like* pada *comment section*, karena seorang *user* dengan tipe pemain *socializer* merasa senang dapat memberikan apresiasi pada *user* lain yang memberikan komen yang dianggap membantu. Dari kumpulan *like* yang diperoleh pada suatu komen, *user* akan mendapatkan *hint point*.
- *Invite user*, karena seorang *user* dengan tipe pemain *socializer* merasa senang dapat berinteraksi dengan lebih banyak *user*. Sehingga *user* dengan tipe ini akan mengundang orang-orang yang belum bergabung untuk dapat bergabung dan berinteraksi di sistem yang sama. *User* yang berhasil mengundang *user* lain untuk bergabung dan memenuhi kondisi akan mendapat *hint point* begitu pula dengan *user* yang diundang.
- *Badges*, karena seorang *user* dengan tipe pemain *achiever* suka mengumpulkan sesuatu yang dapat ditunjukkan kepada *user* lain.
- *Point* yang diwujudkan dalam bentuk XP, karena seorang *achiever* menyukai naik *level* dengan cepat sehingga tipe pemain ini senang mengumpulkan XP sebanyak-banyaknya agar dapat segera naik ke *level* berikutnya.
- *Leaderboard*, karena seorang *user* dengan tipe pemain *achiever* merasa senang apabila berada pada urutan pertama di *leaderboard* untuk kepentingan statusnya.

f) *Deploy the appropriate tools*

Sistem yang akan di-deploy berbasis web dimana tampilan dari sistem ini akan dibuat lebih *responsive* sehingga dapat diakses dengan mudah menggunakan perangkat *computer* maupun *mobile*. *Feedback* yang dapat diperoleh *user* yaitu berupa notifikasi, selain itu juga *user* dapat memperoleh *reward* berupa poin XP

jika berhasil menyelesaikan soal dengan benar dan dapat memperoleh *reward badges* maupun poin XP jika berhasil menyelesaikan *achievement*.

Pada saat *user* membuka UMN OJ, maka *user* harus melakukan login. Jika *user* belum mempunyai akun, maka *user* dapat membuat akun terlebih dahulu. *User* yang baru membuat akun akan berada pada *level* 1. Pada UMN OJ terdapat soal-soal yang dapat dikerjakan oleh *user*. Soal-soal tersebut memiliki minimum *required level* untuk dapat dikerjakan. Jika berhasil mengerjakan soal, maka *user* akan mendapat poin berupa XP. Jika *user* telah mengumpulkan jumlah XP tertentu maka *user* akan naik *level*.

Terdapat *Achievement* yang dapat diselesaikan oleh *user*, dan kemudian akan mendapatkan *reward* berupa *Badges*. Terdapat *Leaderboard* yang akan berisi 10 *user* terbaik yang diurutkan berdasarkan *level* tertinggi, jumlah XP terbanyak, dan jumlah *badges* terbanyak.

Selain itu terdapat *quest* dimana *user* akan mendapatkan soal baru yang dapat dikerjakan dalam batas waktu tertentu untuk mendapatkan bonus XP. Terdapat beberapa kondisi *achievement* yang dapat diselesaikan *user* seperti mencapai *level* tertentu, menyelesaikan *quest* berturut-turut, dan lain-lain. *User* juga dapat memberikan petunjuk pada soal yang ada untuk *user* lain yang kesulitan melalui fitur komentar. Elemen *game* yang akan diterapkan antara lain:

Elemen *dynamics* yang dapat diwujudkan adalah sebagai berikut.

1. *Constraint*: batasan-batasan dalam sistem ini adalah sebagai berikut.
 - XP hanya diperoleh *user* sekali saat *user* berhasil mengerjakan suatu soal. Jika *user* mengerjakan soal yang sama dengan cara berbeda dan benar, maka *user* tidak akan memperoleh XP lagi untuk soal tersebut.

- *Badges* hanya diperoleh *user* sekali saat *user* berhasil menyelesaikan *achievement, quest*, atau memenuhi kondisi tertentu yang telah ditentukan.
 - Jumlah karakter yang ada pada kolom komentar dibatasi sebanyak seratus enam puluh karakter untuk mencegah kecurangan.
2. *Emotions*: emosi yang mendorong keterlibatan *user* dengan permainan adalah sebagai berikut.
- *User* merasa senang menemukan *hidden badge*.
 - *User* merasa senang berhasil mengerjakan soal dengan benar.
 - *User* merasa berkompetisi untuk masuk ke *leaderboard*.
 - *User* merasa senang dapat berinteraksi lewat kolom komentar.
 - *User* merasa senang dapat mengundang orang untuk dapat bergabung dan berinteraksi di sistem.
 - *User* memiliki rasa kebanggaan menjadi yang pertama berhasil menyelesaikan tantangan yang ada.
3. *Progression*: perkembangan *user* yang dicatat oleh sistem adalah sebagai berikut.
- Mencatat jumlah soal yang telah berhasil dikerjakan *user* dari total soal yang ada.
 - Mencatat jumlah XP yang telah diperoleh *user* sejak awal *user* bergabung dalam sistem.
 - Menampilkan jumlah XP yang dibutuhkan *user* untuk naik ke *level* selanjutnya.
 - Mencatat jumlah *badge* yang telah diperoleh *user*.

4. *Relationship*: interaksi sosial yang dapat dilakukan antar *user* adalah sebagai berikut.

- *User* dapat melempar pertanyaan terkait soal yang ada, seperti mengkonfirmasi *test case*.
- *User* dapat berbagi informasi tambahan berupa *test case* tambahan terkait soal yang ada lewat kolom komentar.
- *User* dapat mengapresiasi komen dari *user* lain yang dianggap bermanfaat atau membantu dengan memberi *like* pada komen tersebut.
- Admin dapat memberikan informasi terkait soal lewat kolom komentar.

Elemen *mechanics* yang akan diterapkan adalah sebagai berikut.

1. *Challenges*: tantangan yang diwujudkan dalam bentuk *quests*.
2. *Feedback*: diwujudkan dalam bentuk notifikasi yang berisi informasi yang diperoleh setelah mengerjakan soal, apabila mendapat XP, *badge* atau naik *level*.
3. *Reward*: diwujudkan dengan mendapatkan XP, *badges*, dan *content unlocking*.
4. *Competitions*: diwujudkan dengan persaingan masuk ke leaderboards.
5. *Resource acquisition*: diwujudkan dengan *badges* untuk dikoleksi dan untuk *badges* tertentu akan mendapatkan soal baru dengan XP yang tinggi.

Elemen *components* yang akan diterapkan adalah sebagai berikut.

1. *Points*: poin yang akan diterapkan berupa *experience point* (XP).
2. *Levels*: *level* yang akan diterapkan berupa *level user*. *Level* dimulai dari 1 dan akan naik saat *user* telah berhasil mengumpulkan XP tertentu.

3. *Content Unlocking*: soal-soal yang ada akan memiliki *minimum required level* untuk dapat dikerjakan dan terdapat soal yang hanya ditampilkan jika *user* memiliki *badges* tertentu.
4. *Leaderboards*: menampilkan 10 *user* dengan level tertinggi, XP terbanyak, dan jumlah *badges* terbanyak.
5. *Quests*: soal yang muncul di jangka waktu tertentu, jika dikerjakan pada jangka waktu yang ditentukan maka akan mendapatkan bonus XP.
6. *Achievements*: kondisi-kondisi tertentu untuk dilakukan oleh *user* agar mendapatkan *reward* adalah sebagai berikut.
 - Pertama kali bergabung di UMN OJ
 - Pertama kali berhasil mengerjakan soal dengan benar.
 - Telah berhasil melewati materi *input output*, iterasi dan kondisi.
 - *User* yang pertama kali berhasil mengerjakan *quest*.
 - Berhasil mengerjakan tiga dari lima *quest*, mendapatkan *hidden badge*.
 - Berhasil melewati materi-materi dasar.
 - Berhasil mengerjakan dua puluh soal *first attempt* akumulasi.
 - Berhasil mengerjakan lima belas soal *first attempt* berturut-turut.
 - Pertama kali berhasil mengerjakan soal dengan kategori materi *Graph*.
 - Pertama kali berhasil mengerjakan soal dengan kategori materi *String*.
 - Pertama kali berhasil mengerjakan soal dengan kategori materi *Math*.
 - Berhasil mengerjakan sepuluh soal.
 - Berhasil mengerjakan dua puluh lima soal.
 - Berhasil mengerjakan lima puluh soal.
 - Berhasil mengerjakan minimal satu soal per hari dalam satu minggu.

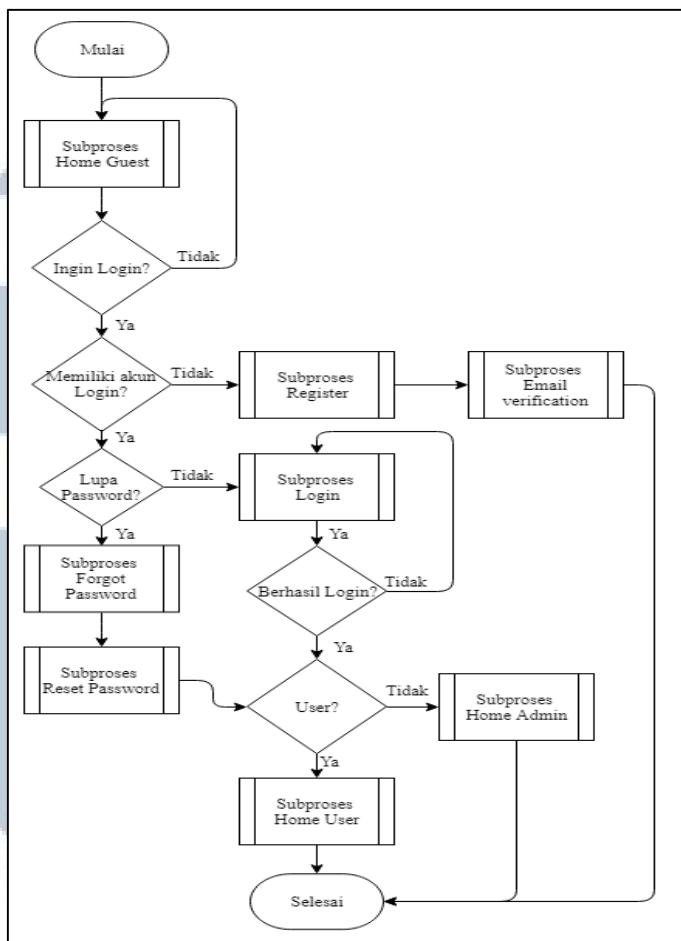
- Berhasil mengerjakan soal setelah lebih dari atau sama dengan lima kali gagal, mendapatkan *hidden badge*.
 - Memiliki tiga *hidden badge*.
 - Sepuluh kali *submit* dan gagal mengerjakan soal yang sama.
 - Aktif dalam kolom komentar di dua belas soal.
 - Memiliki total XP lebih dari atau sama dengan tujuh belas ribu.
7. *Badges*: representasi visual yang didapatkan setelah menyelesaikan *achievements*.
8. *Collections*: daftar kumpulan *badges* yang dapat diperoleh *user*.
9. *Gifting*: *hint point* yang dapat diperoleh dengan *like* pada *comment section* di soal untuk mengapresiasi komen dari *user* lain dan *invite user*. *User* yang mendapat *like* dengan jumlah tertentu pada komennya akan mendapatkan *hint point*. *User* yang meng-*invite* orang untuk bergabung di UMN OJ jika memenuhi kondisi *user* tersebut serta *user* yang di-*invite* akan mendapat *hint point*. *Hint point* dapat digunakan untuk membuka *hint* yang ada disoal.

3.3 Perancangan Sistem

Metode perancangan yang digunakan pada penelitian ini adalah metode prosedural, sehingga dalam proses perancangan digunakan *flowchart*, *Entity Relationship Diagram* (ERD), *database schema*, struktur tabel dan *design user interface*.

3.3.1 Flowchart

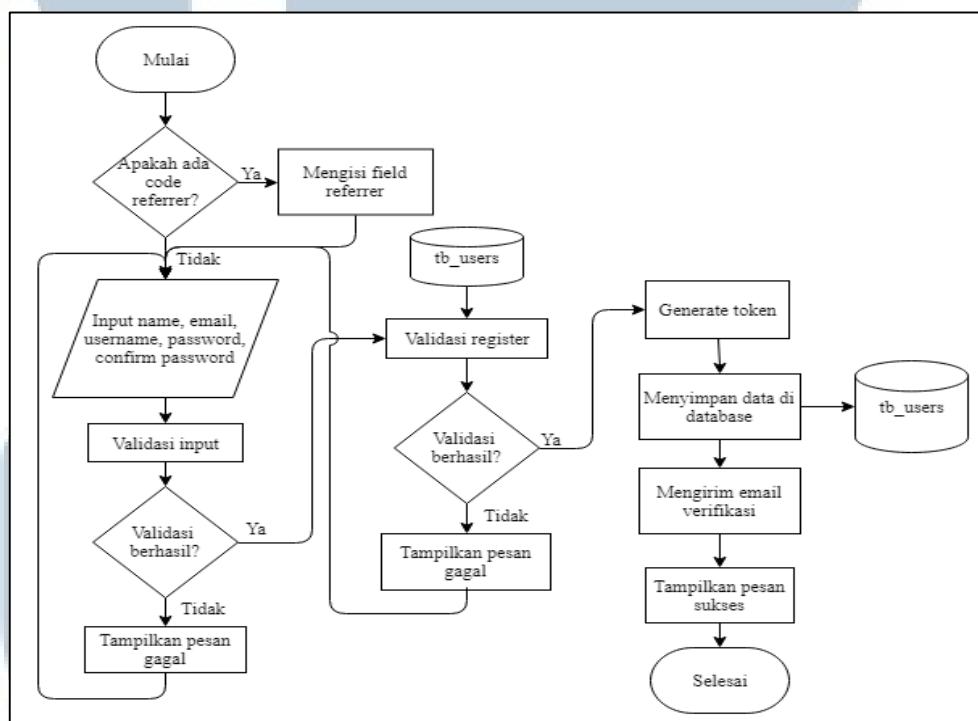
Flowchart ditujukan untuk menggambarkan bagaimana proses gamifikasi UMN *online judge*.



Gambar 3.1 Flowchart Gamifikasi UMN Online Judge

Gambar 3.1 menunjukkan *flowchart* dari proses gamifikasi UMN *online judge*. Pada proses ini, pertama kali akan masuk ke subproses Home Guest, lalu mengecek apakah *user* ingin melakukan *login*. Jika *user* tidak ingin *login*, maka *user* akan tetap berada di subproses Home Guest yang akan dijelaskan lebih *detail* pada Gambar 3.5. Jika *user* ingin *login*, maka dilakukan konfirmasi apakah *user* sudah memiliki akun atau belum. Jika *user* belum memiliki akun, maka *user* akan masuk ke subproses Register yang akan dijelaskan lebih *detail* pada Gambar 3.2 kemudian lanjut masuk ke subproses Email Verification yang akan dijelaskan lebih *detail* pada Gambar 3.3. Jika *user* sudah memiliki akun, maka *user* akan dilakukan konfirmasi apakah *user* lupa password atau tidak. Jika tidak, maka *user* masuk ke subproses Login yang akan dijelaskan lebih *detail* pada Gambar 3.4. Jika ya, maka

user akan masuk ke subproses Forgot Password yang akan dijelaskan lebih *detail* pada Gambar 3.32 kemudian lanjut masuk ke subproses Reset Password yang akan dijelaskan lebih *detail* pada Gambar 3.33. Jika *user* tidak berhasil *login*, maka *user* akan kembali ke subproses Login. Jika *user* berhasil *login*, maka akan dilakukan pengecekan apakah *user* yang login merupakan *user* biasa atau admin. Jika *user* yang *login* merupakan *user* biasa, maka *user* akan masuk ke subproses Home User yang akan dijelaskan lebih *detail* pada Gambar 3.10. Jika *user* yang *login* merupakan admin, maka *user* akan masuk ke subproses Home Admin yang akan dijelaskan lebih *detail* pada Gambar 3.23.

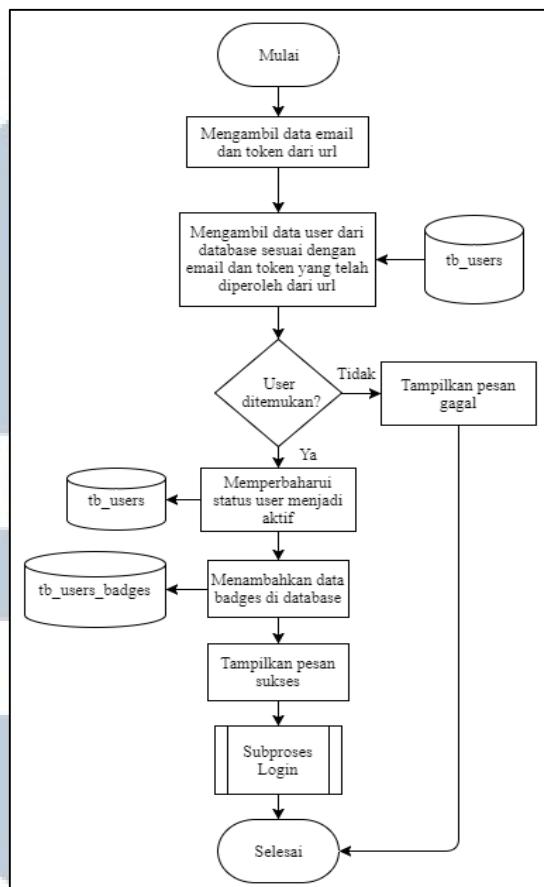


Gambar 3.2 Flowchart Register

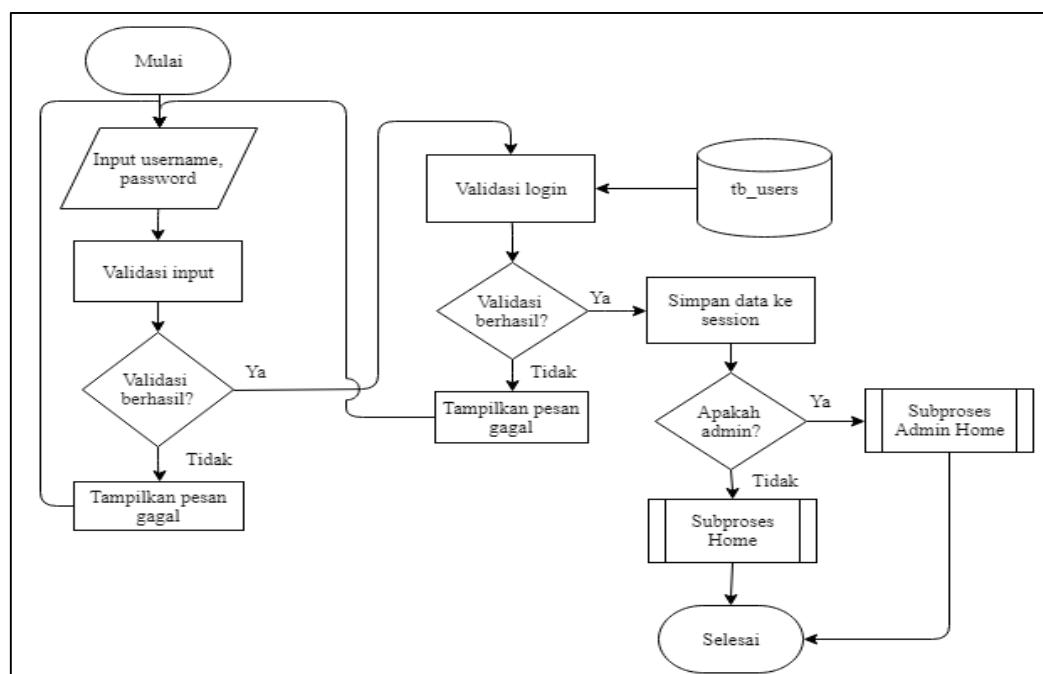
Gambar 3.2 menunjukkan *flowchart* dari proses *register*. Pada saat melakukan register akan dilakukan pengecekan apakah terdapat *code referrer* terlebih dahulu. Jika ada, maka sistem akan mengisi *field code referrer* secara otomatis. Setelah itu *user* memasukkan *name*, *email*, *username*, *password* dan *confirm password*. Jika tidak ada, maka *user* dapat langsung memasukkan data di

atas. Kemudian, dari data yang telah dimasukkan sebelumnya akan dilakukan validasi yaitu dengan mengecek apakah semua *field* telah diisi, selain itu akan dicek juga *field email* menggunakan format *email* dan *confirm password* sama dengan *password*. Jika validasi gagal, maka akan ditampilkan pesan gagal dan *user* akan diminta untuk memasukkan kembali data pada *field* yang tersedia sesuai dengan format yang benar. Jika validasi berhasil, maka akan dilakukan validasi *register* yaitu dengan mengecek apakah *email* atau *username* yang dimasukkan sudah pernah digunakan atau belum. Jika validasi gagal, maka akan ditampilkan pesan gagal dan *user* akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka akan dilakukan *generate token* dan data yang telah dimasukkan beserta token akan disimpan di *database* lalu akan dikirimkan *email* untuk diverifikasi oleh *user* dan akan ditampilkan pesan sukses.

Gambar 3.3 menunjukkan *flowchart* dari proses *email verification*. Pada proses ini, dilakukan pengambilan data *email* dan *token* dari *url*. Lalu mengambil data *user* dari *database* yang sesuai dengan *email* dan *token* yang diperoleh dari *url*. Kemudian dilakukan pengecekan, apakah terdapat data *user* di *database* dengan *email* dan *token* tersebut. Jika tidak ada, maka tampilkan pesan gagal. Jika ada, maka akan dilakukan pembaharuan data status *user* di *database* menjadi aktif dan menambahkan data *badges* di *database* untuk *user* yang pertama kali bergabung. Setelah itu, akan ditampilkan pesan sukses dan *user* akan masuk ke subproses Login.



Gambar 3.3 Flowchart Email Verification

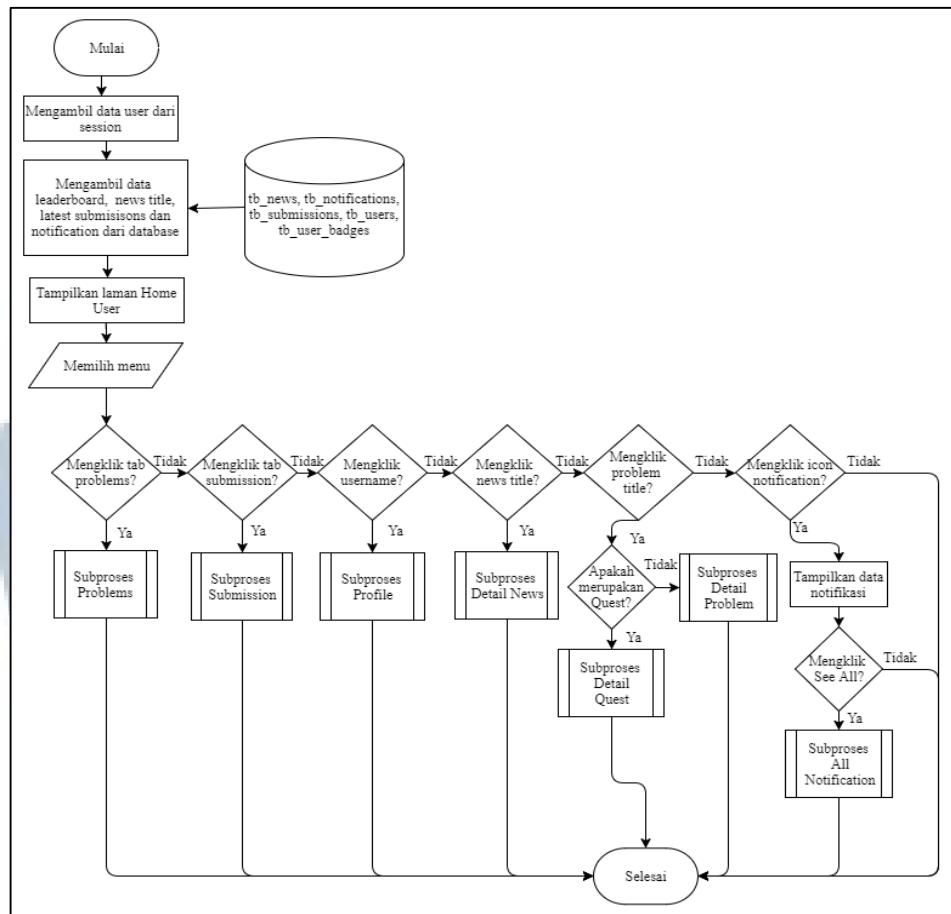


Gambar 3.4 Flowchart Login

N U S A N T A R A

Gambar 3.4 menunjukkan *flowchart* dari proses *login*. Untuk melakukan *login user* harus memasukkan *username* dan *password*. Kemudian, dari data yang telah dimasukkan sebelumnya akan dilakukan validasi yaitu dengan mengecek apakah semua *field* telah diisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan *user* akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka akan dilakukan validasi *login* yaitu dengan mengecek apakah *user* dengan *username* dan *password* tersebut terdapat datanya di *database* dan merupakan *user* aktif atau tidak. Jika validasi gagal, maka akan ditampilkan pesan gagal dan *user* akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data *user* akan disimpan ke *session* dan akan masuk ke subproses Home.

Gambar 3.5 menunjukkan *flowchart* dari proses *home guest*. Pada proses ini dilakukan pengambilan data *leaderboard* berupa data *level*, *username*, *xp*, dan *badges*. Lalu mengambil data *latest submission* berupa data *problem title*, *username*, *verdict*, dan *submission time* dari *database* kemudian ditampilkan di laman Home Guest. Jika *guest* mengklik tab problems, maka *guest* akan masuk ke subproses Problems Guest. Jika *guest* mengklik tab submissions, maka *guest* akan masuk ke subproses Submissions Guest. Jika *guest* mengklik *username*, maka *guest* akan masuk ke subproses Profile User. Jika *guest* mengklik tab about, maka *guest* akan masuk ke tampilan laman About. Jika *guest* mengklik *problem title* dari *problem* yang telah dipilih, maka *guest* akan masuk ke subproses Detail Problem Guest. Jika *guest* mengklik salah satu *news title*, maka *guest* akan masuk ke subproses Detail News.

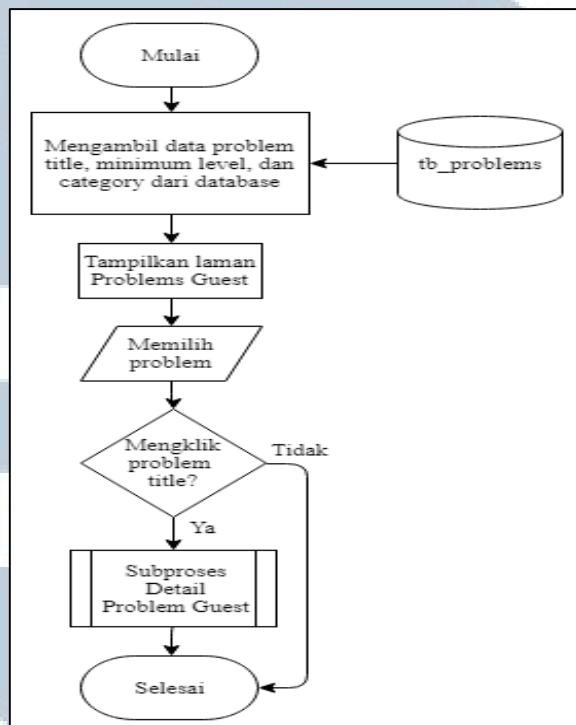


Gambar 3.5 Flowchart Home Guest

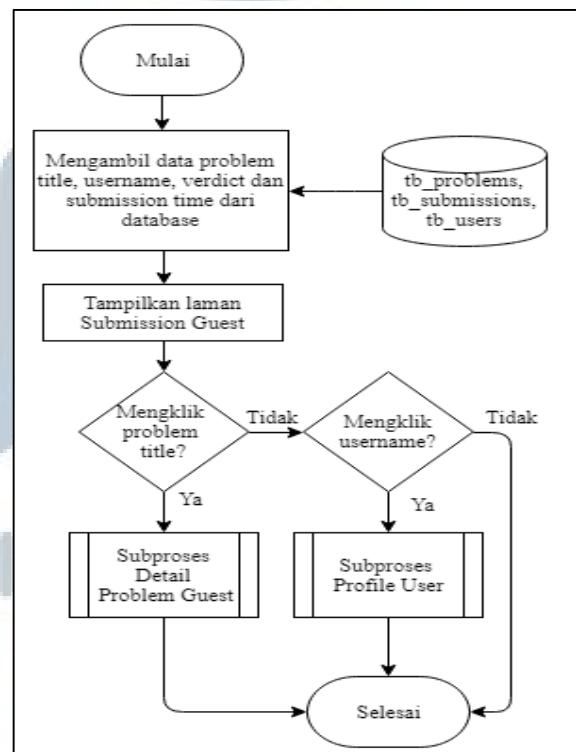
Gambar 3.6 menunjukkan *flowchart* dari proses *problems guest*. Pada proses ini dilakukan pengambilan data *problem title*, minimum level, dan *category* dari *database* kemudian ditampilkan di laman *Problems Guest*. Setelah melihat daftar *problem*, *guest* memilih salah satu *problem* yang ada. Jika *guest* mengklik *problem title* dari *problem* yang telah dipilih, maka *guest* akan masuk ke subproses *Detail Problem Guest*.

Gambar 3.7 menunjukkan *flowchart* dari proses *submissions guest*. Pada proses ini dilakukan pengambilan data *problem title*, *username*, *verdict*, dan *submission time* dari *database* kemudian ditampilkan di laman *Submission Guest*. Jika *guest* mengklik *problem title* dari *problem* yang telah dipilih, maka *guest* akan

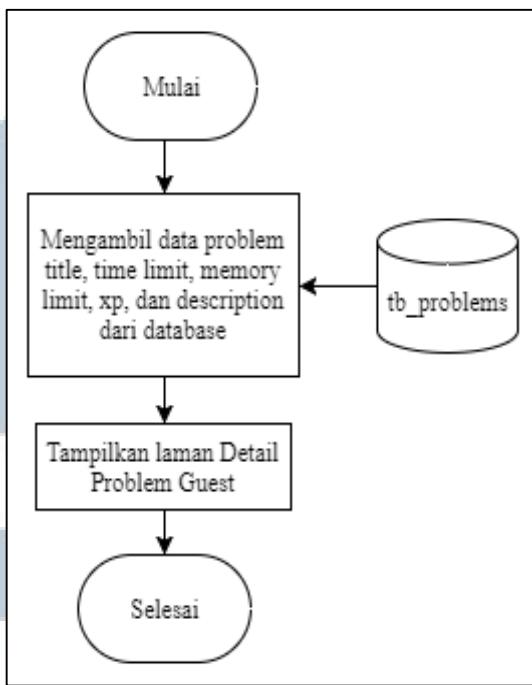
masuk ke subproses Detail Problem Guest. Jika *user* mengklik *username*, maka *user* akan masuk ke subproses Profile User.



Gambar 3.6 Flowchart Problems Guest



Gambar 3.7 Flowchart Submissions Guest

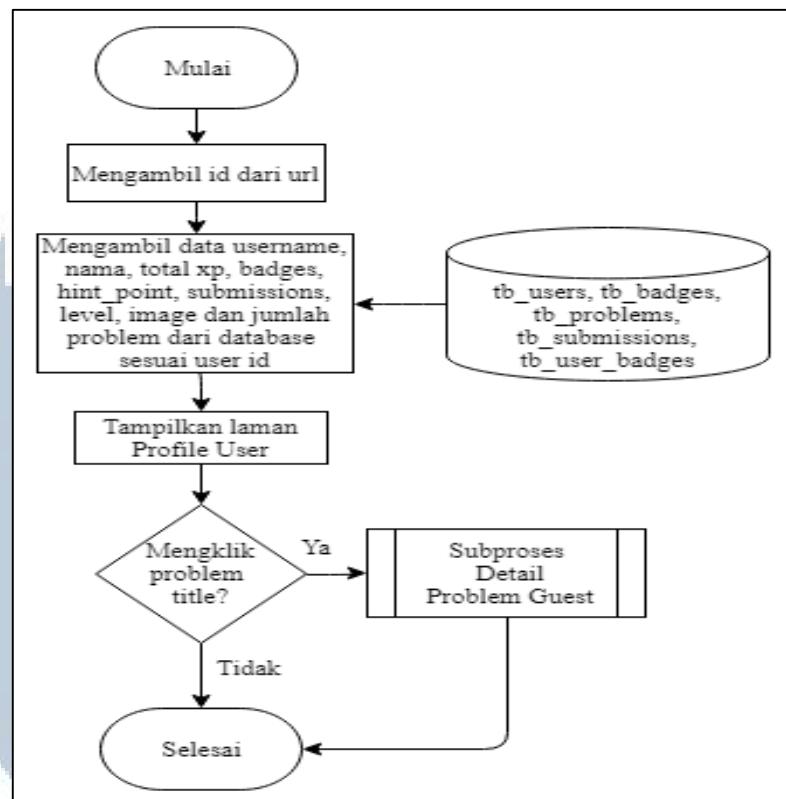


Gambar 3.8 Flowchart Detail Problem Guest

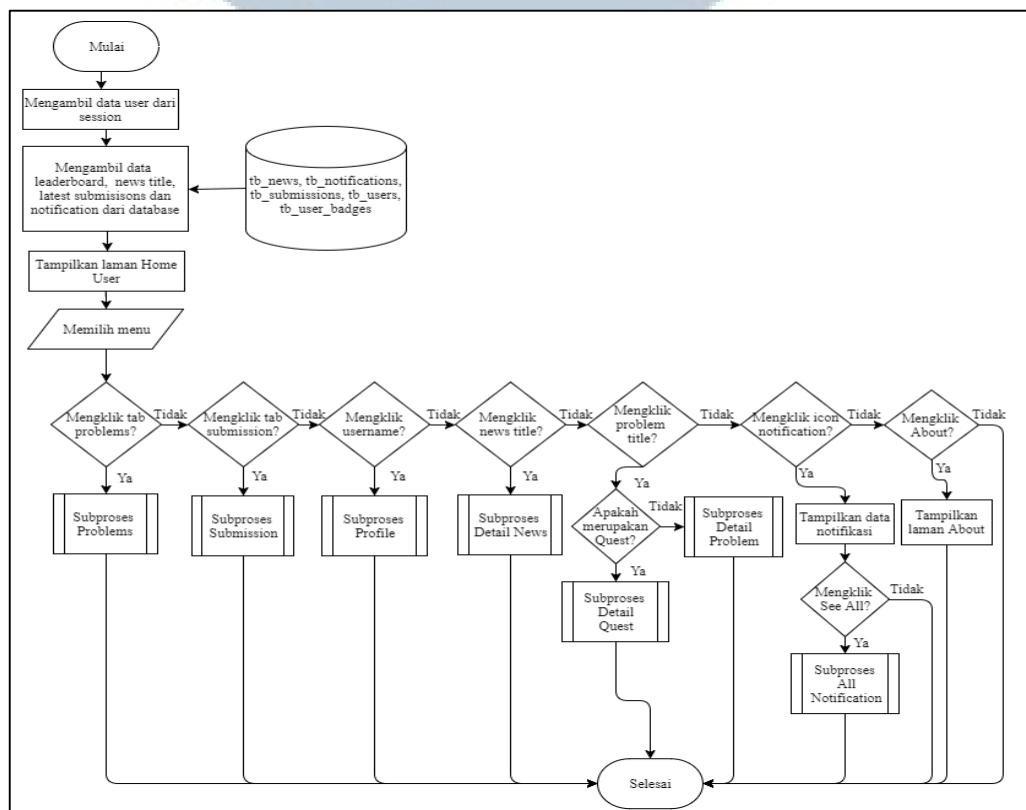
Gambar 3.8 menunjukkan *flowchart* dari proses *detail problem guest*. Pada proses ini dilakukan pengambilan data *problem title*, *time limit*, *memory limit*, *xp*, dan *description* dari *database* kemudian ditampilkan di laman *Detail Problem Guest*.

Gambar 3.9 menunjukkan *flowchart* dari proses *profile user*. Pada proses ini dilakukan pengambilan data *id* dari *url*, lalu mengambil data *username*, *name*, *total xp*, *badges*, *hint_point*, *level*, *image*, jumlah *problem*, dan *submissions* berupa data *problem title*, dan *verdict* dari *database* sesuai *user id* kemudian ditampilkan di laman *Profile User*. Jika *guest* mengklik *problem title* dari *problem* yang telah dipilih, maka *guest* akan masuk ke subproses *Detail Problem Guest*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.9 Flowchart Profile User

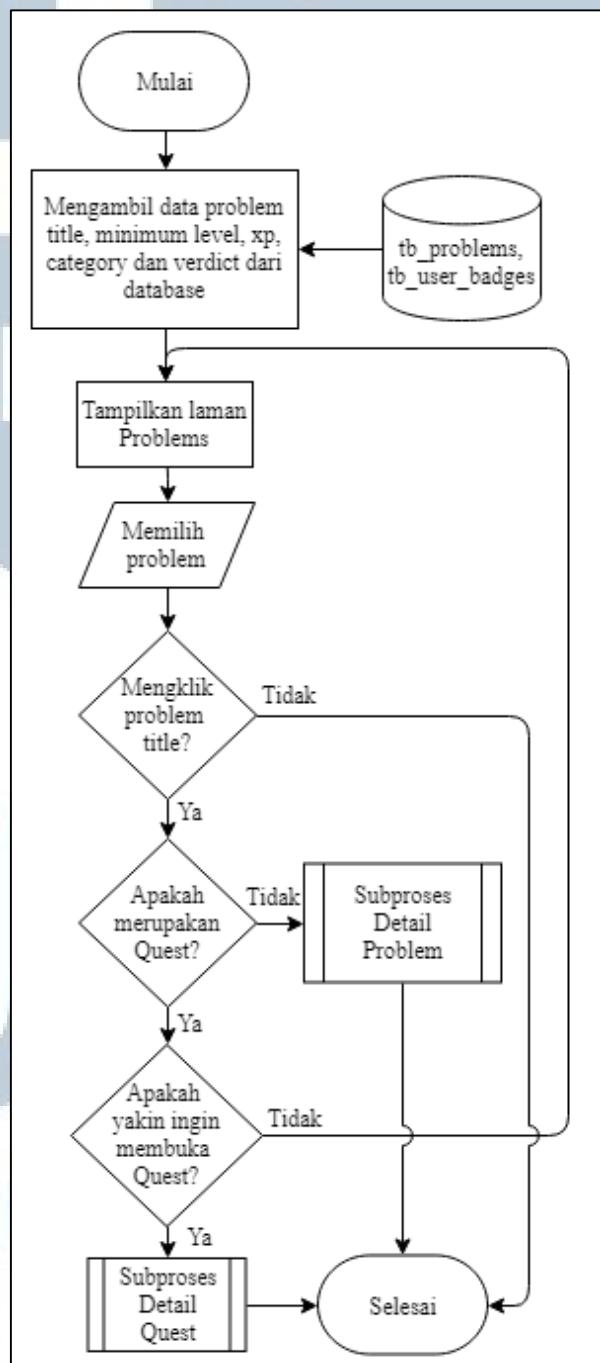


N U S A N T A R A
Gambar 3.10 Flowchart Home User

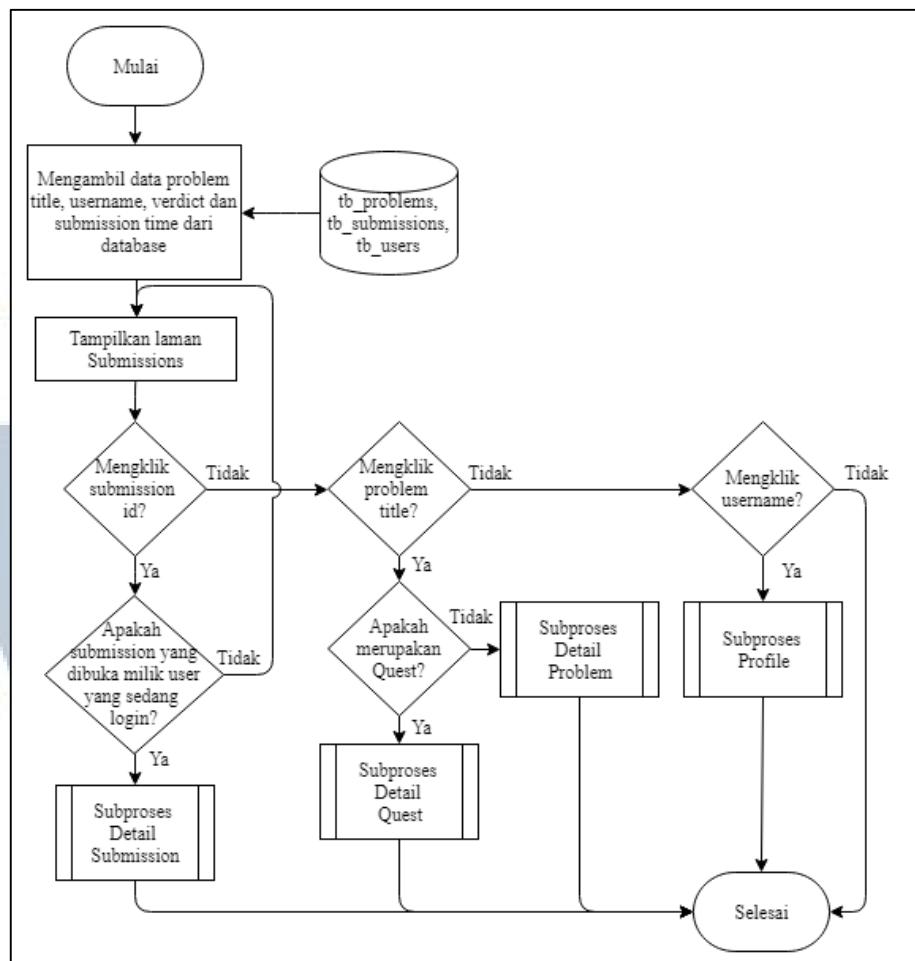
Gambar 3.10 menunjukkan *flowchart* dari proses *home user*. Pada proses ini dilakukan pengambilan data *user* dari *session* lalu mengambil data *leaderboard* berupa data *level*, *username*, *xp*, dan *badges*. Lalu mengambil data *news title*, dan *latest submission* berupa data *problem title*, *username*, *verdict*, dan *submission time* dari *database* kemudian ditampilkan di laman Home User. Jika *user* mengklik tab problems, maka *user* akan masuk ke subproses Problems. Jika *user* mengklik tab submissions, maka *user* akan masuk ke subproses Submissions. Jika *user* mengklik username, maka *user* akan masuk ke subproses Profile. Jika *user* mengklik salah satu dari news title, maka *user* akan masuk ke subproses Detail News. Jika *user* mengklik *problem title* dari *problem* yang telah dipilih, maka akan dilakukan pengecekan apakah *problem title* yang diklik merupakan *problem* biasa atau *quest*. Jika *problem* tersebut merupakan *problem* biasa, maka *user* akan masuk ke subproses Detail Problem. Jika *problem* tersebut merupakan *problem* *quest*, maka *user* akan masuk ke subproses Detail Quest. Jika *user* mengklik *icon notification*, maka akan ditampilkan data notifikasi yang masuk. Jika *user* mengklik See All pada notifikasi, maka *user* akan masuk ke subproses All Notification. Jika *user* mengklik tab about, maka *user* akan masuk ke tampilan laman About.

Gambar 3.11 menunjukkan *flowchart* dari proses *problems*. Pada proses ini dilakukan pengambilan data *problem title*, *xp*, *minimum level*, *category* dan *verdict* dari *database* kemudian ditampilkan di laman Problems. Setelah melihat daftar *problem*, *user* memilih salah satu *problem* yang ada. Jika *user* mengklik *problem title* dari *problem* yang telah dipilih, maka akan dilakukan pengecekan apakah *problem title* yang diklik merupakan *problem* biasa atau *quest*. Jika *problem* tersebut merupakan *problem* biasa, maka *user* akan masuk ke subproses Detail

Problem. Jika *problem* tersebut merupakan *problem quest*, maka akan muncul *modal dialog* untuk konfirmasi apakah *user* akan membuka *quest* tersebut atau tidak. Jika ya, maka *user* akan masuk ke subproses Detail Quest dan *timer* otomatis akan berjalan sesuai durasi waktu penggerjaan yang telah ditetapkan.



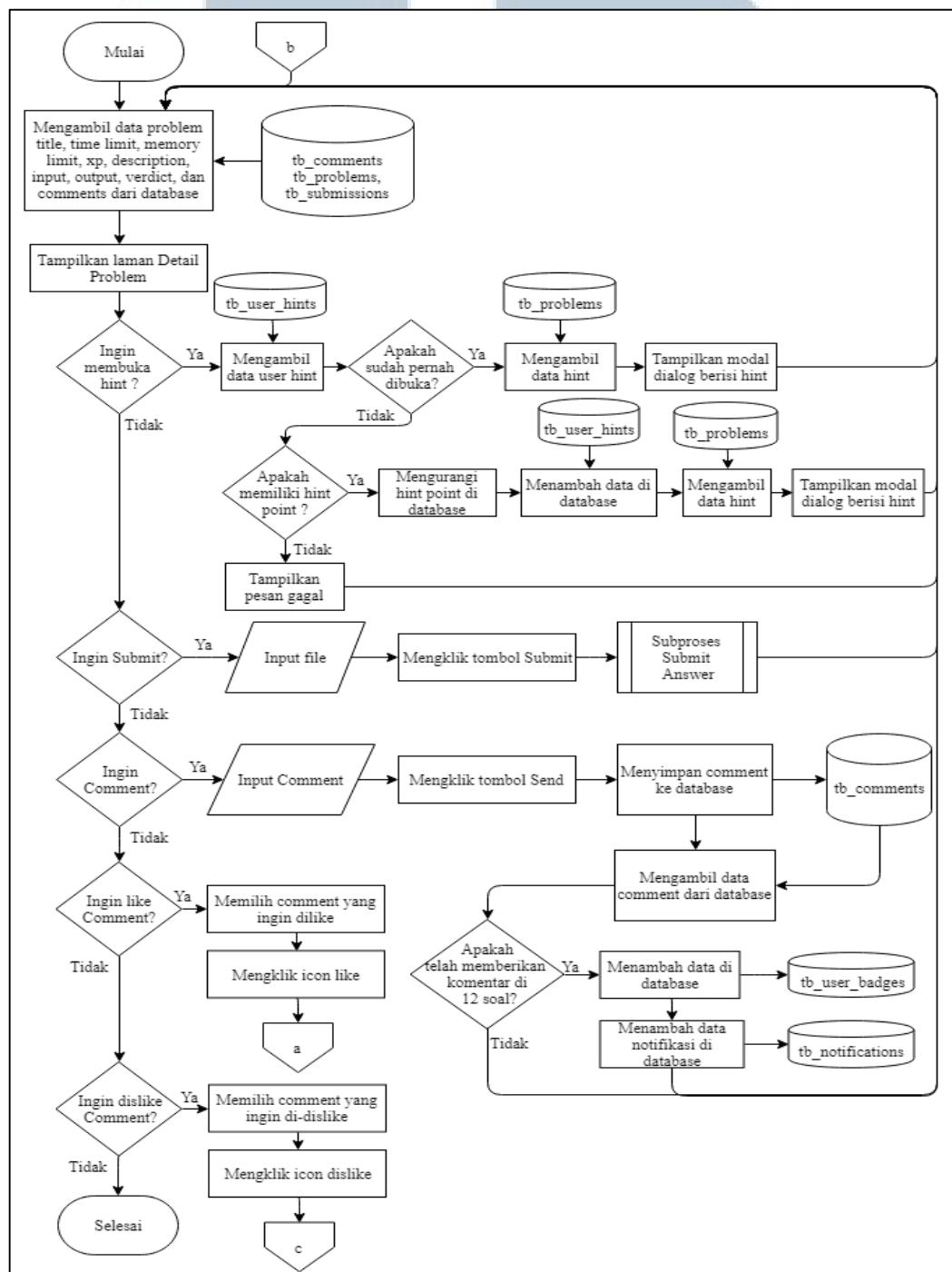
Gambar 3.11 Flowchart Problems



Gambar 3.12 Flowchart Submissions

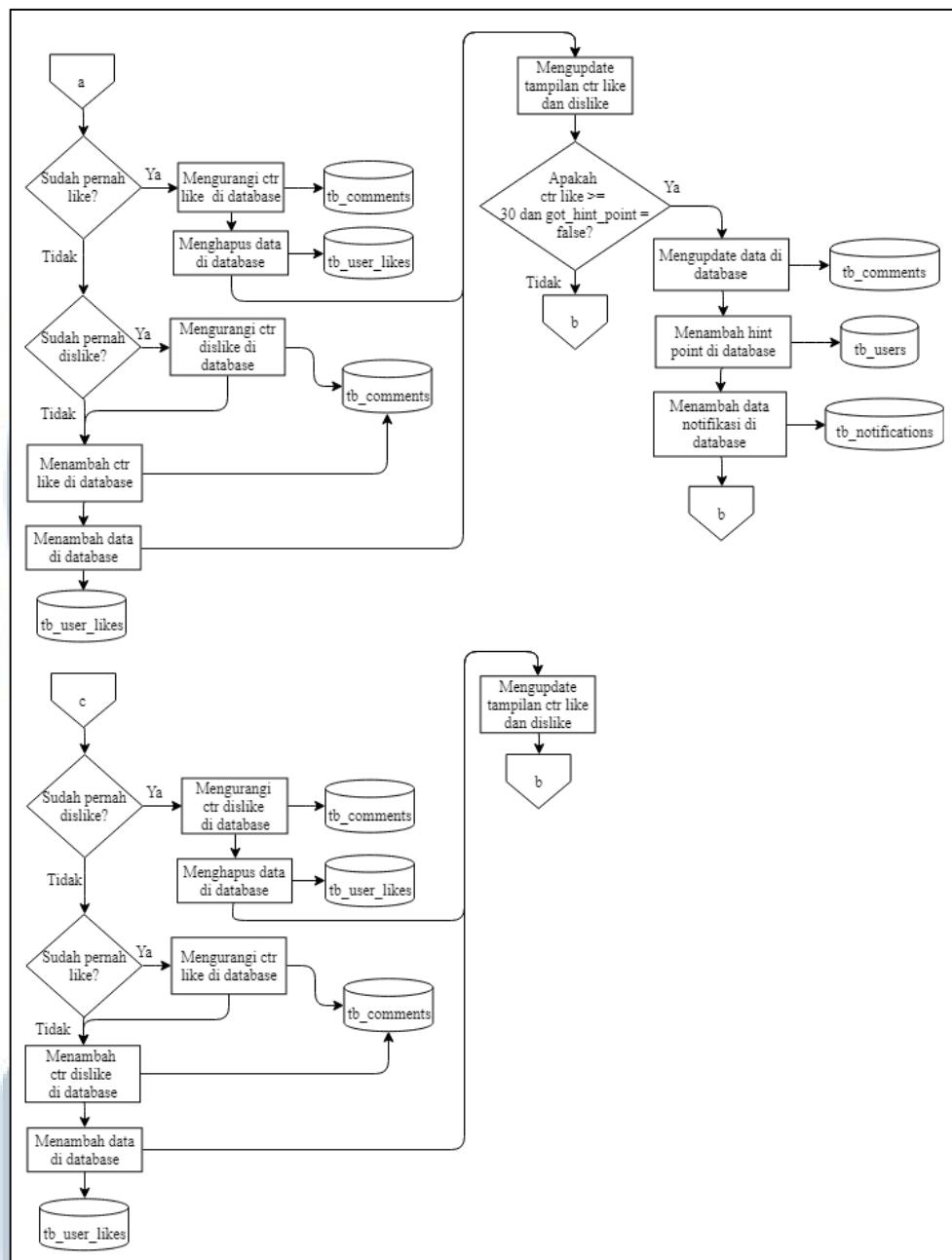
Gambar 3.12 menunjukkan *flowchart* dari proses *submissions*. Pada proses ini dilakukan pengambilan data *problem title*, *username*, *verdict*, dan *submission time* dari *database* kemudian ditampilkan di laman *Submissions*. Jika *user* mengklik *submission id* dari *problem* yang telah dipilih, maka akan dilakukan pengecekan apakah *submission* yang ingin dibuka merupakan *submission* milik *user* yang sedang *login*. Jika ya, maka akan masuk ke subproses *Detail Submission*. Jika tidak, maka akan tetap berada di tampilan laman *Submissions*. Jika *user* mengklik *problem title* dari *problem* yang telah dipilih, maka akan dilakukan pengecekan apakah *problem title* yang diklik merupakan *problem* biasa atau *quest*. Jika *problem* tersebut merupakan *problem* biasa, maka *user* akan masuk ke

subproses Detail Problem. Jika *problem* tersebut merupakan *problem quest*, maka *user* akan masuk ke subproses Detail Quest. Jika *user* mengklik *username*, maka *user* akan masuk ke subproses Profile.



Gambar 3.13 Flowchart Detail Problem

N U S A N T A R A



Gambar 3.13 Flowchart Detail Problem (Lanjutan)

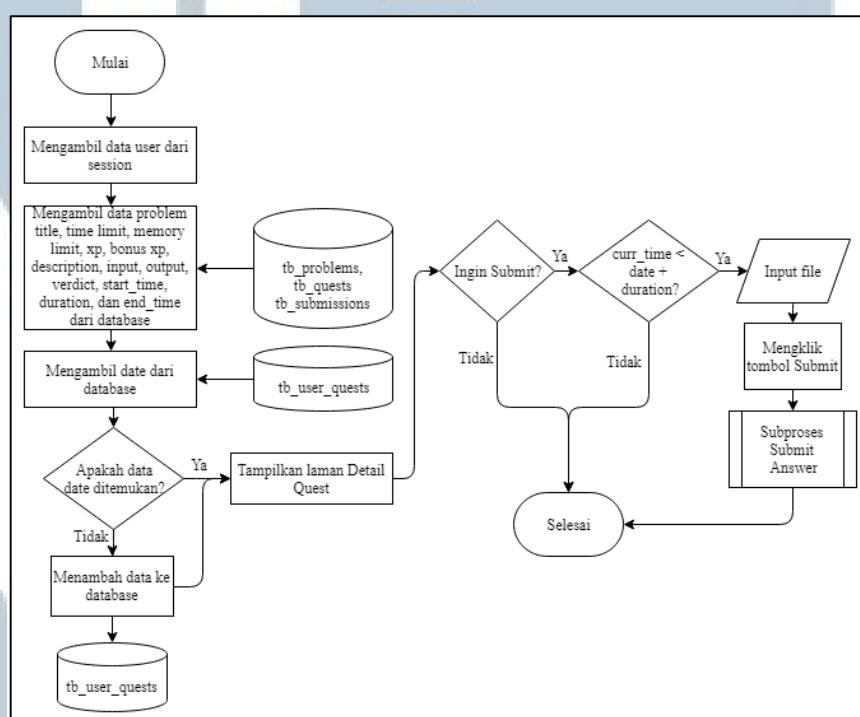
Gambar 3.13 menunjukkan *flowchart* dari proses *detail problem*. Pada proses ini dilakukan pengambilan data *problem title*, *time limit*, *memory limit*, *xp*, *description*, *input*, *output*, *verdict*, dan *comments* dari *database* kemudian ditampilkan di laman Detail Problem. Jika *user* ingin membuka *hint*, maka dilakukan pengecekan apakah *user* tersebut sudah pernah membuka *hint* tersebut. Jika *user* sudah pernah membuka *hint* tersebut, maka ditampilkan *hint* dari *problem*

tersebut. Jika *user* belum pernah membuka *hint* tersebut, maka dilakukan pengecekan apakah *hint point* yang dimiliki *user* cukup. Jika cukup, maka poin *user* akan dikurangi, kemudian *hint* dari *problem* ditampilkan. Namun, jika poin tidak cukup, maka akan ditampilkan pesan gagal.

Jika *user* ingin melakukan *submit*, maka *user* dapat mengunggah *file* jawaban pada *field* yang tersedia lalu mengklik tombol Submit dan *user* akan masuk ke subproses Submit Answer. Jika *user* ingin memberi komen, maka *user* dapat mengetikkan komen pada *field* yang tersedia lalu mengklik tombol Send dan data komen tersebut akan disimpan di *database*. Lalu akan dilakukan pengambilan data komen dari tabel comments. Setelah itu dilakukan pengecekan apakah *user* telah memberikan komentar di dua belas soal berbeda. Jika ya, maka akan menambah data di tabel user_badges dan di tabel notifications. Jika tidak, maka akan dilakukan pengambilan data *problem title*, *time limit*, *memory limit*, *xp*, *description*, *input*, *output*, *verdict*, dan *comments* dari *database* kemudian ditampilkan di laman Detail Problem.

Jika *user* ingin men-like suatu *comment*, maka *user* dapat menekan ikon *like*. Jika *user* sudah pernah *like* pada *comment* tersebut, maka ctr_like pada tabel comments akan dikurangi, kemudian data pada tabel user_likes akan dihapus. Jika *user* sebelumnya men-dislike *comment* tersebut, maka ctr_dislike pada tabel comments akan dikurangi. Jika *user* belum pernah *like* sebelumnya, maka ctr_like akan ditambah pada tabel comments dan menambah data pada tabel user_likes. Setelah itu, tampilan *counter like* dan *dislike* di-update. Kemudian, jika ctr_like lebih dari 30 dan *user* belum pernah mendapat *hint point* dari *comment* tersebut, maka *user* yang memberi *comment* akan diberi tambahan *hint point*.

Jika *user* ingin men-*dislike* suatu *comment*, maka *user* dapat menekan ikon *dislike*. Jika *user* sudah pernah *dislike* pada *comment* tersebut, maka *ctr_dislike* pada tabel *comments* akan dikurangi, kemudian data pada tabel *user_likes* akan dihapus. Jika *user* sebelumnya men-*like* *comment* tersebut, maka *ctr_like* pada tabel *comments* akan dikurangi. Jika *user* belum pernah *dislike* sebelumnya, maka *ctr_dislike* akan ditambah pada tabel *comments* dan menambah data pada tabel *user_likes*. Setelah itu, tampilan *counter like* dan *dislike* di-*update*.

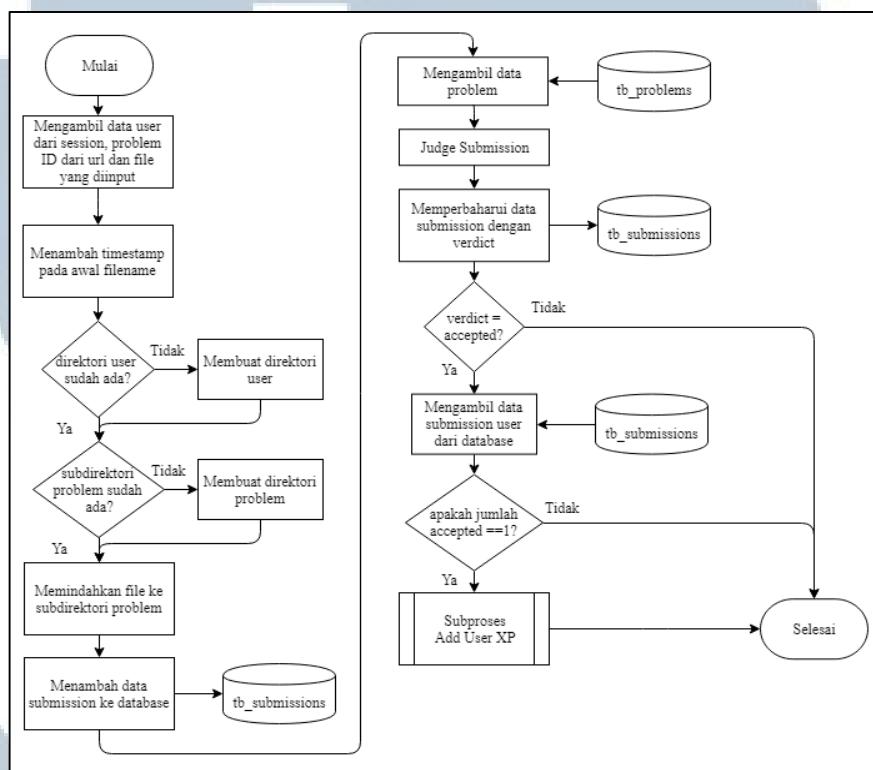


Gambar 3.14 Flowchart Detail Quest

Gambar 3.14 menunjukkan *flowchart* dari proses *detail quest*. Pada proses ini dilakukan pengambilan data *problem title*, *time limit*, *memory limit*, *xp*, *bonus xp*, *description*, *input*, *output*, *verdict*, *start time*, *end time*, dan *duration* dari *database* kemudian mengambil data *date* di *database*. Jika data *date* tidak ditemukan, maka akan ditambah data *date* ke *database*, kemudian menampilkan tampilan laman *Detail Quest*. Jika data *date* ditemukan, maka akan menampilkan tampilan laman *Detail Quest*. Jika *user* ingin *submit*, maka akan dilakukan pengecekan apakah

waktu saat ini kurang dari waktu buka laman Detail Quest (*date*) ditambah durasi.

Jika kurang, maka *user* dapat dapat mengunggah *file* jawaban pada *field* yang tersedia lalu mengklik tombol submit dan *user* akan masuk ke subproses Submit Answer. Jika lebih, maka *user* tidak dapat melakukan submit dan selesai. Jika *user* tidak ingin submit, maka selesai.

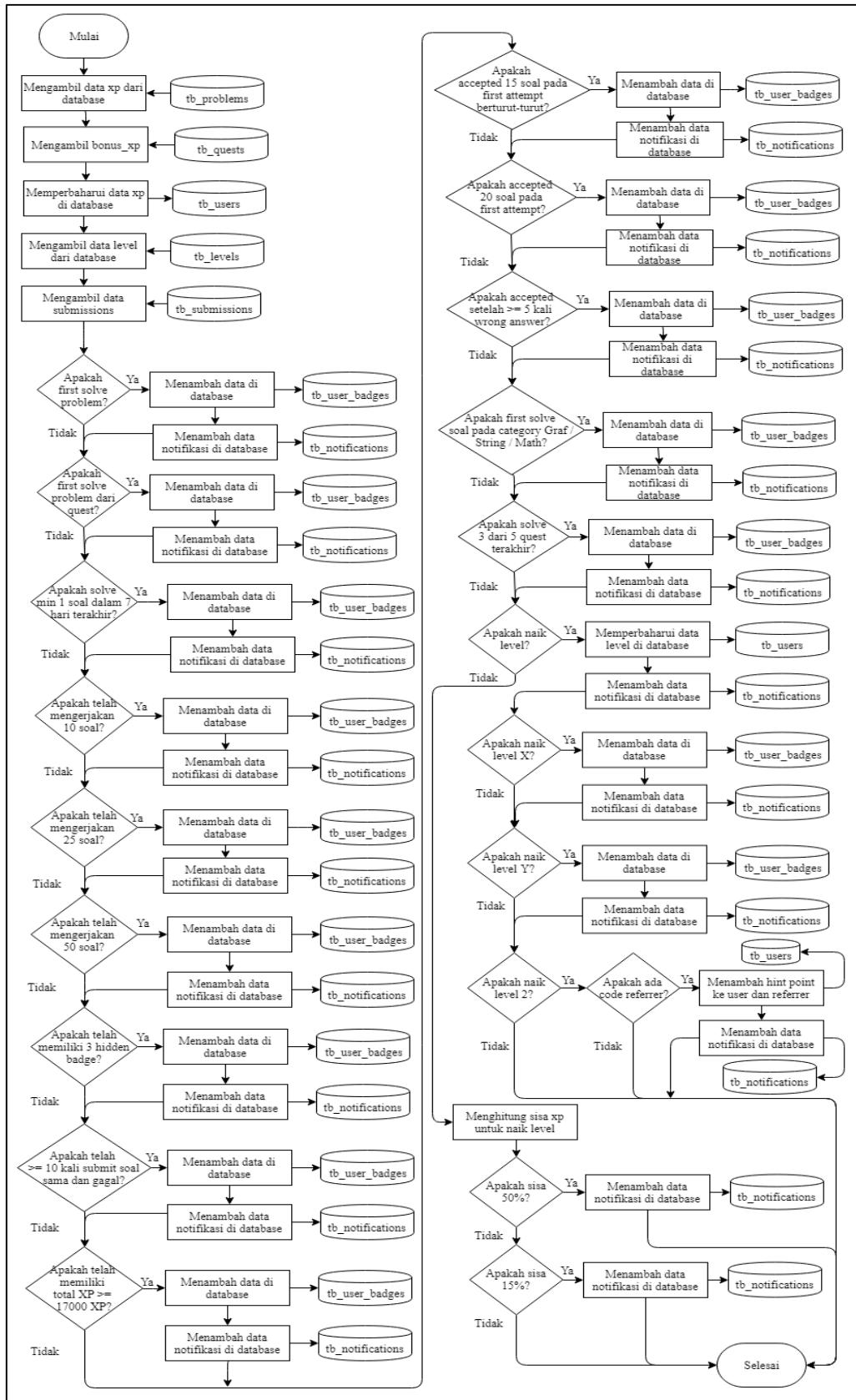


Gambar 3.15 Flowchart Submit Answer

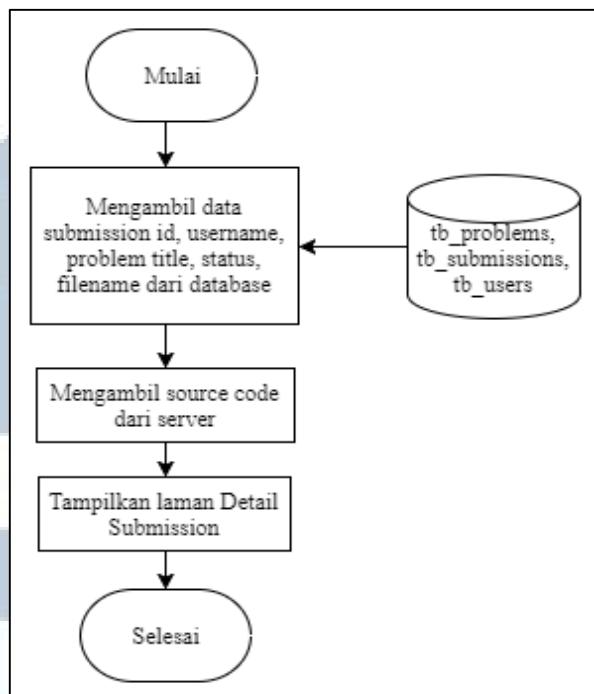
Gambar 3.15 menunjukkan *flowchart* dari proses *submit answer*. Pada proses ini, dilakukan pengambilan data *user* dari *session*, *problem id* dari *url* dan *file* yang di-*input*, kemudian menambahkan *timestamp* pada awal *filename*. Lalu dilakukan pengecekan apakah *user* sudah mempunyai direktori atau belum. Jika belum punya direktori, maka akan membuat direktori untuk *user* terlebih dahulu. Jika sudah punya direktori, maka akan dicek apakah subdirektori untuk *problem* sudah ada atau belum. Jika belum ada subdirektori, maka akan membuat direktori untuk *problem* terlebih dahulu. Jika sudah ada, maka *file* akan dipindahkan ke subdirektori

problem, lalu menambah data *submission* ke tabel *submissions*. Setelah itu, mengambil data *problem* dari tabel *problems* untuk dilakukan *judge submission*. Kemudian memperbarui data *submission* di tabel *submissions* dengan *verdict* yang diperoleh dari hasil *judge*. Lalu dilakukan pengecekan apakah *verdict* yang diperoleh *accepted* atau tidak. Jika *accepted*, maka akan dilakukan pengambilan data *submission user* dari tabel *submissions*. Setelah itu, dilakukan pengecekan apakah jumlah *accepted* sama dengan satu. Jika ya, maka akan masuk ke subproses Add User XP. Jika tidak, maka pengecekan selesai.

Gambar 3.16 menunjukkan *flowchart* dari proses *add user xp*. Pada proses ini, dilakukan pengambilan data *xp* dari tabel *problems* data *bonus xp* dari tabel *quests*, lalu memperbarui data *xp* di tabel *users*. Setelah itu, mengambil data *level* dari tabel *levels* dan mengambil data *submissions* dari tabel *submissions*. Kemudian dilakukan pengecekan apakah *user* memenuhi kondisi untuk mendapatkan *badges*. Jika kondisi terpenuhi, maka dilakukan penambahan data pada tabel *user_badges* dan tabel *notifications*. Selain itu, dilakukan pengecekan apakah *xp user* memenuhi batas *xp* untuk naik ke *level* selanjutnya. Jika *user* naik *level*, maka dilakukan pembaharuan data *level* di tabel *users* serta menambah data notifikasi di tabel *notifications*. Jika tidak, maka dilakukan pengecekan juga apakah sisa *xp* yang diperlukan *user* untuk naik *level* sisa lima belas persen atau dua puluh lima persen. Jika ya, maka akan menambah data notifikasi di *database*. Dilakukan pengecekan juga apakah *user* naik ke *level* dua. Jika ya, maka akan dicek apakah *user* memiliki *code referrer*. Jika ya, maka *user* dan *referrer* akan mendapat *hint point*. Data notifikasi juga akan ditambahkan ke *database*. Jika tidak, maka pengecekan selesai.



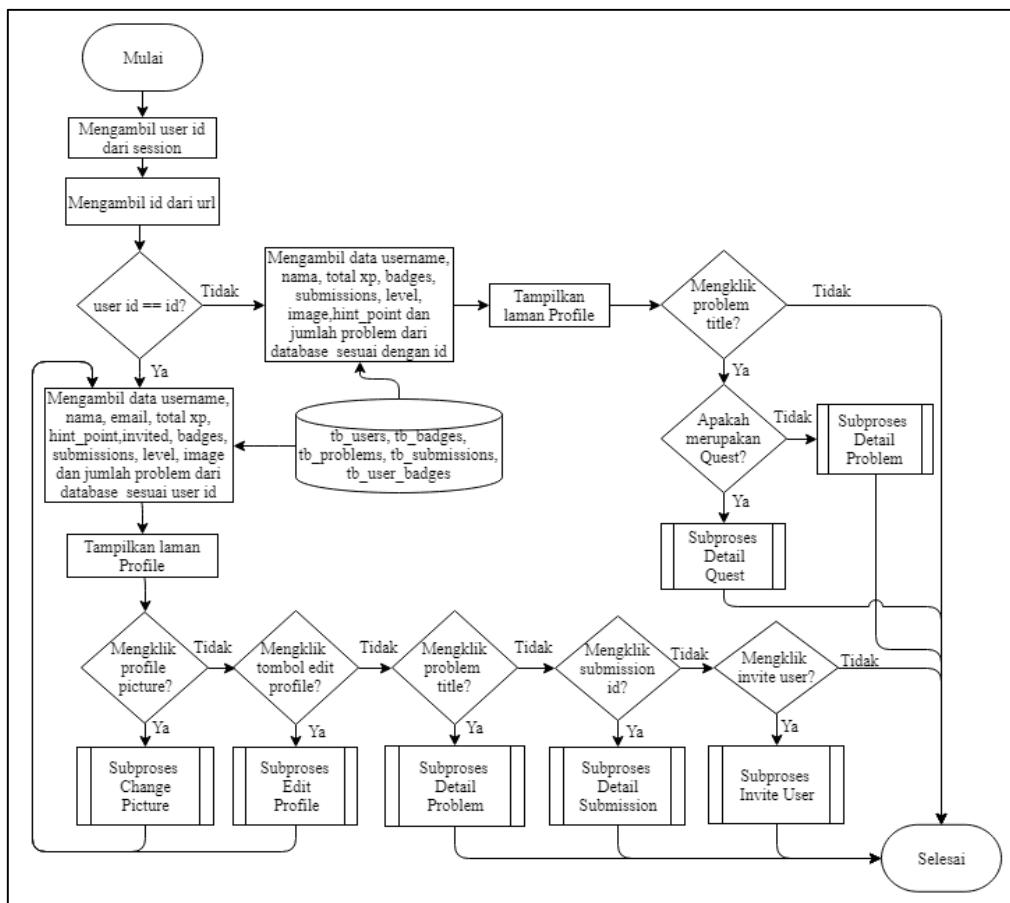
Gambar 3.16 Flowchart Add User XP



Gambar 3.17 Flowchart Detail Submission

Gambar 3.17 menunjukkan *flowchart* dari proses *detail submission*. Pada proses ini dilakukan pengambilan data *submission id*, *username*, *problem title*, *status* dan *filename* jawaban dari *database* lalu mengambil *source code* dari *server* berdasarkan *filename* kemudian ditampilkan di laman *Detail Submission*.

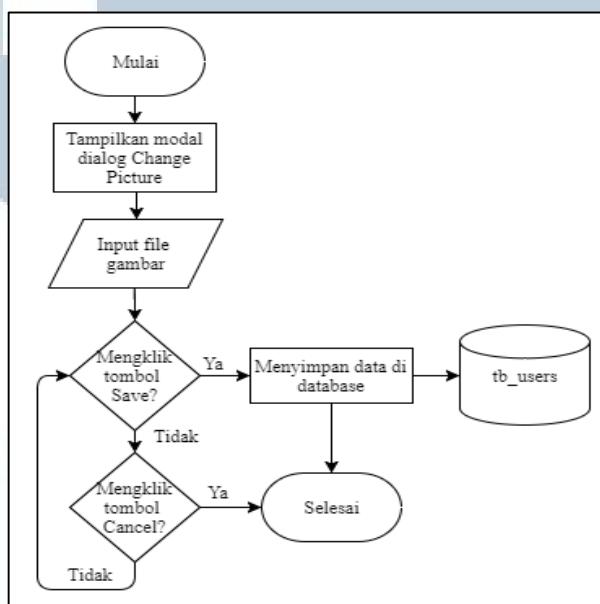




Gambar 3.18 Flowchart Profile

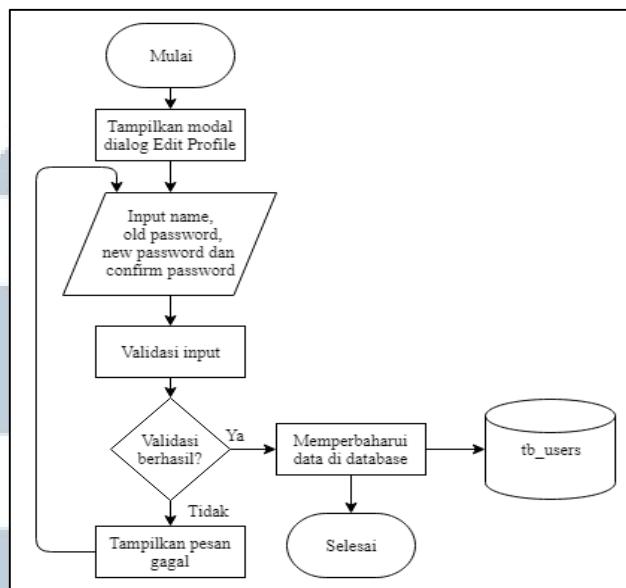
Gambar 3.18 menunjukkan *flowchart* dari proses *profile*. Pada proses ini dilakukan pengambilan data *user* dari *session* lalu mengambil data *id* dari *url*. Jika *user id* sekarang tidak sama dengan *id* yang diperoleh dari *url*, maka akan dilakukan pengambilan data *username*, *name*, jumlah *user* yang telah diundang, *total xp*, *badges*, *level*, jumlah *problem*, dan *submissions* berupa data *problem title*, dan *verdict* dari *database* sesuai *user id* kemudian ditampilkan di laman *Profile*. Jika *user* mengklik *problem title*, maka akan dilakukan pengecekan apakah *problem title* yang diklik merupakan *problem biasa* atau *quest*. Jika *problem* tersebut merupakan *problem biasa*, maka *user* akan masuk ke subproses *Detail Problem*. Jika *problem* tersebut merupakan *problem quest*, maka *user* akan masuk ke subproses *Detail Quest*. Jika *user id* sekarang sama dengan *id* yang diperoleh dari *url*, maka akan

pengambilan data *username*, *name*, *email*, *total xp*, jumlah *user* yang telah diundang, jumlah *hint point*, *badges*, *level*, jumlah *problem*, *submissions* berupa data *submission id*, *problem title*, dan *verdict* dari *database* sesuai *user id* kemudian ditampilkan di laman Profile. Jika *user* mengklik *problem title*, maka *user* akan masuk ke subproses Detail Problem. Jika *user* mengklik *submission id*, maka *user* akan masuk ke subproses Detail Submission. Jika *user* mengklik *profile picture*, maka *user* akan masuk ke subproses Change Picture. Jika *user* mengklik tombol edit profile, maka *user* akan masuk ke subproses Edit Profile. Jika *user* mengklik tombol invite user, maka *user* akan masuk ke subproses Invite User.



Gambar 3.19 Flowchart Change Picture

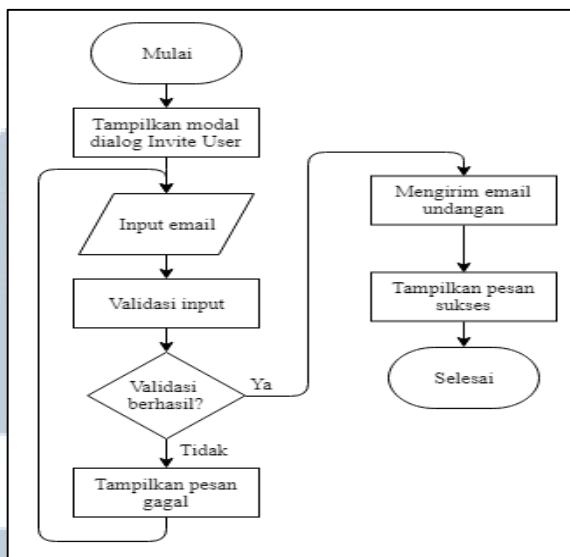
Gambar 3.19 menunjukkan *flowchart* dari proses *change picture*. Pada proses ini akan menampilkan *modal dialog* dan *user* dapat mengunggah *file* gambar yang akan dijadikan foto profil baru. Jika *user* mengklik tombol *Save*, maka data *file* gambar tadi akan disimpan di *database*. Jika *user* mengklik tombol *Cancel*, maka *modal dialog* tersebut akan tertutup.



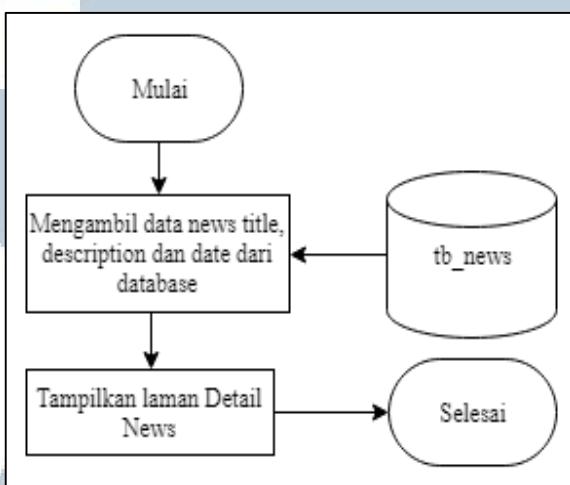
Gambar 3.20 Flowchart Edit Profile

Gambar 3.20 menunjukkan *flowchart* dari proses *edit profile*. Pada proses ini akan menampilkan *modal dialog* dan *user* dapat memasukkan data *name*, *old password*, *new password*, dan *confirm password*. Setelah itu, dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi dan *confirm password* dan *password* sama. Jika validasi gagal, maka akan ditampilkan pesan gagal dan *user* akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data di *database* akan diperbaharui dengan data yang baru dan *modal dialog* akan tertutup.

Gambar 3.21 menunjukkan *flowchart* dari proses *invite user*. Pada proses ini akan menampilkan *modal dialog* dan *user* dapat memasukkan data *email* yang akan diundang. Setelah itu akan dilakukan validasi dari *email* tersebut apakah sesuai format *email* atau tidak. Jika validasi berhasil, maka sistem akan mengirim *email* undangan ke alamat *email* yang telah dimasukkan sebelumnya. Kemudian akan ditampilkan pesan sukses. Jika validasi gagal, maka akan ditampilkan pesan gagal dan meminta *user* untuk memasukkan kembali alamat *email*.



Gambar 3.21 Flowchart Invite User



Gambar 3.22 Flowchart Detail News

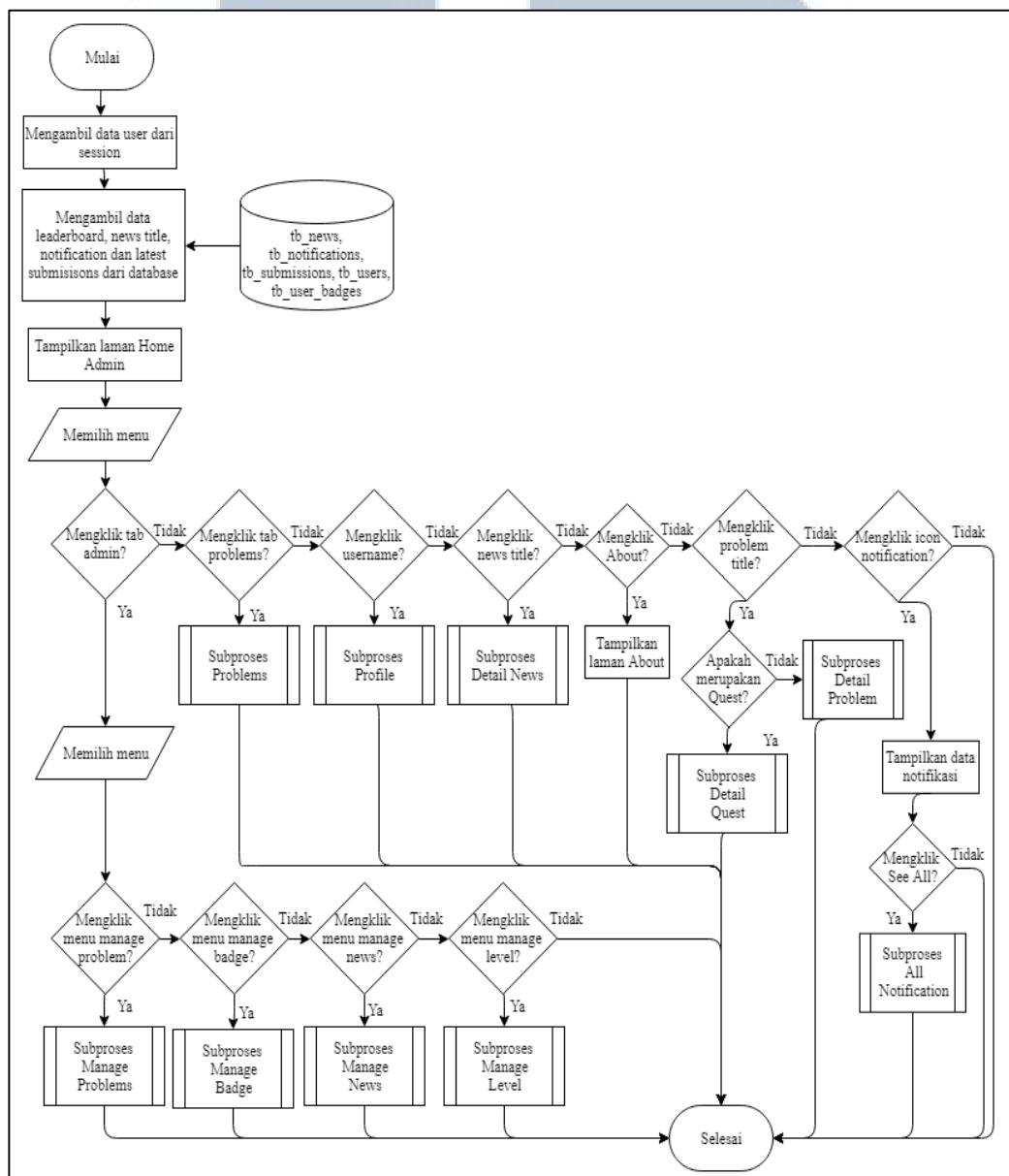
Gambar 3.22 menunjukkan *flowchart* dari proses *detail news*. Pada proses ini dilakukan pengambilan data *news title*, *description*, dan *date* dari *database* kemudian ditampilkan di laman *Detail News*.

Gambar 3.23 menunjukkan *flowchart* dari proses *home admin*. Pada proses ini dilakukan pengambilan data *user* dari *session* lalu mengambil data *leaderboard* berupa data *level*, *username*, *xp*, dan *badges*. Kemudian mengambil data *news title*, dan *latest submission* berupa data *problem title*, *username*, *verdict*, dan *submission time* dari *database* kemudian ditampilkan di laman *Home Admin*. Setelah itu,

admin memilih salah satu menu yang ada. Jika admin mengklik tab admin, lalu admin memilih menu manage problem dan mengklik menu tersebut, maka admin akan masuk ke subproses Manage Problems. Jika admin memilih menu manage badge dan mengklik menu tersebut, maka admin akan masuk ke subproses Manage Badge. Jika admin memilih menu manage news dan mengklik menu tersebut, maka admin akan masuk ke subproses Manage News. Jika admin memilih menu manage level dan mengklik menu tersebut, maka admin akan masuk ke subproses Manage Level. Jika admin mengklik tab problems, maka admin akan masuk ke subproses Problems. Jika admin mengklik *username*, maka *user* akan masuk ke subproses Profile. Jika admin mengklik salah satu dari news title, maka admin akan masuk ke subproses Detail News. Jika admin mengklik tab about, maka admin akan masuk ke tampilan laman About. Jika admin mengklik *problem title* dari *problem* yang telah dipilih, maka akan dilakukan pengecekan apakah *problem title* yang diklik merupakan *problem* biasa atau *quest*. Jika *problem* tersebut merupakan *problem* biasa, maka admin akan masuk ke subproses Detail Problem. Jika *problem* tersebut merupakan *problem quest*, maka admin akan masuk ke subproses Detail Quest. Jika admin mengklik *icon notification*, maka akan ditampilkan data notifikasi yang masuk. Jika admin mengklik See All pada notifikasi, maka admin akan masuk ke subproses All Notification.

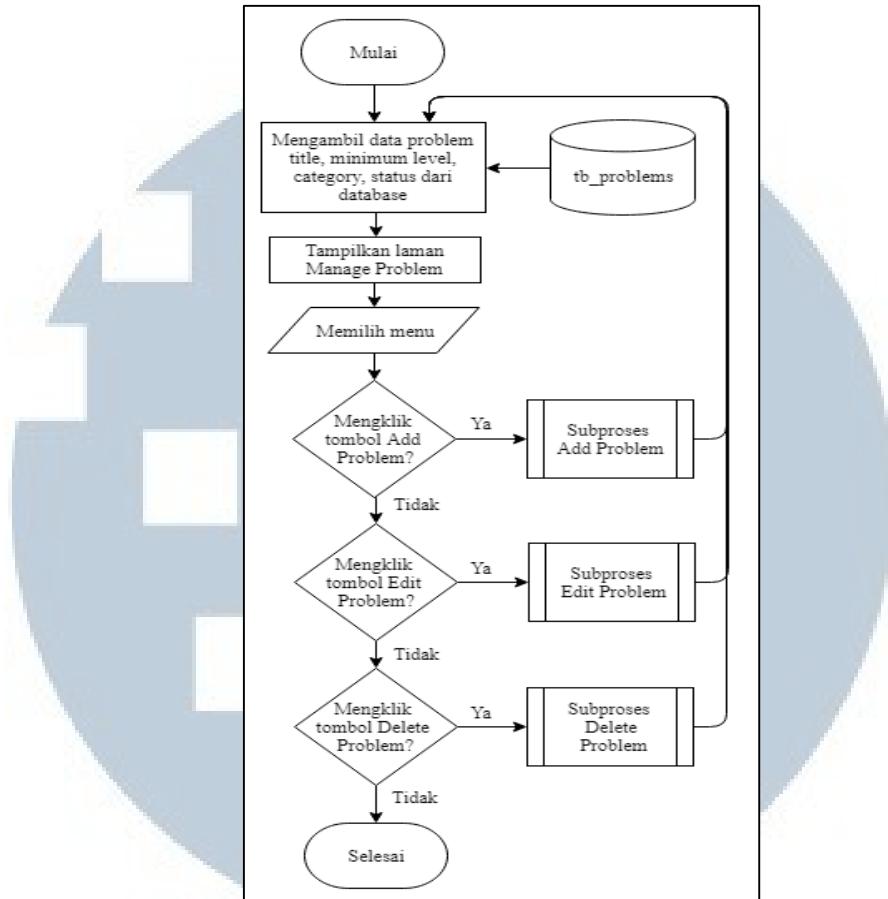
Gambar 3.24 menunjukkan *flowchart* dari proses *manage problem*. Pada proses ini akan dilakukan pengambilan data *problem title*, *minimum level*, *category*, dan *status* dari *database* kemudian ditampilkan di laman Manage Problem. Setelah itu, admin memilih salah satu menu yang ada. Jika admin mengklik tombol Add Problem, maka admin akan masuk ke subproses Add Problem. Jika admin mengklik

tombol Edit Problem, maka admin akan masuk ke subproses Edit Problem. Jika admin mengklik tombol Delete Problem, maka admin akan masuk ke subproses Delete Problem.

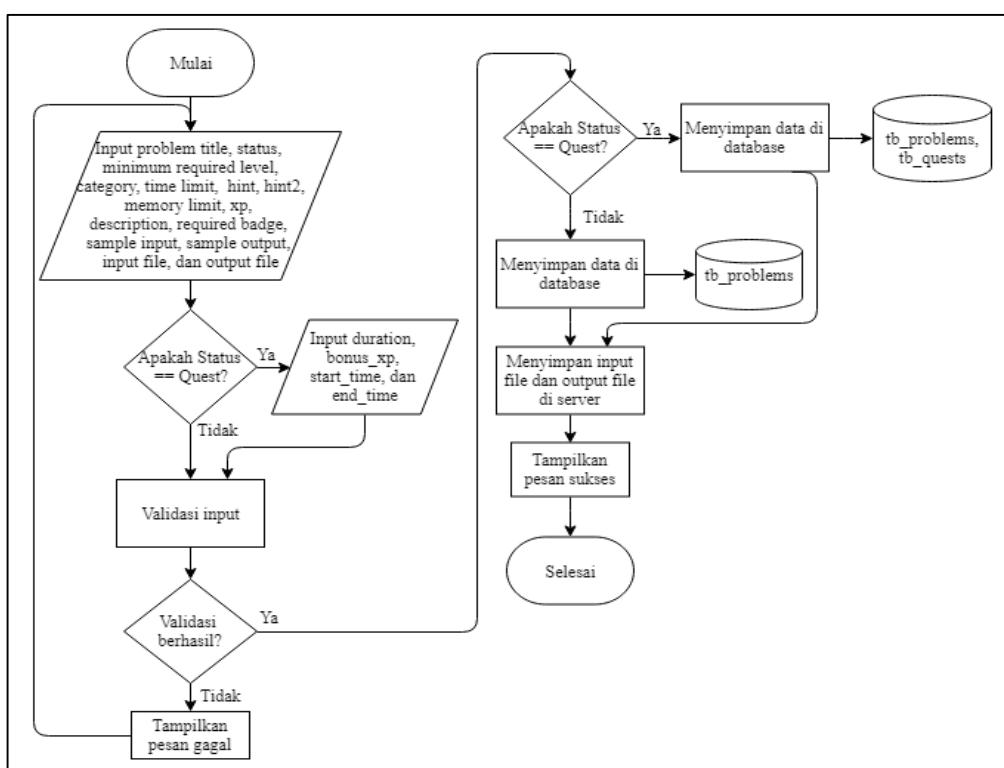


Gambar 3.23 Flowchart Home Admin

UNIVERSITAS
MULTIMEDIA
NUSANTARA



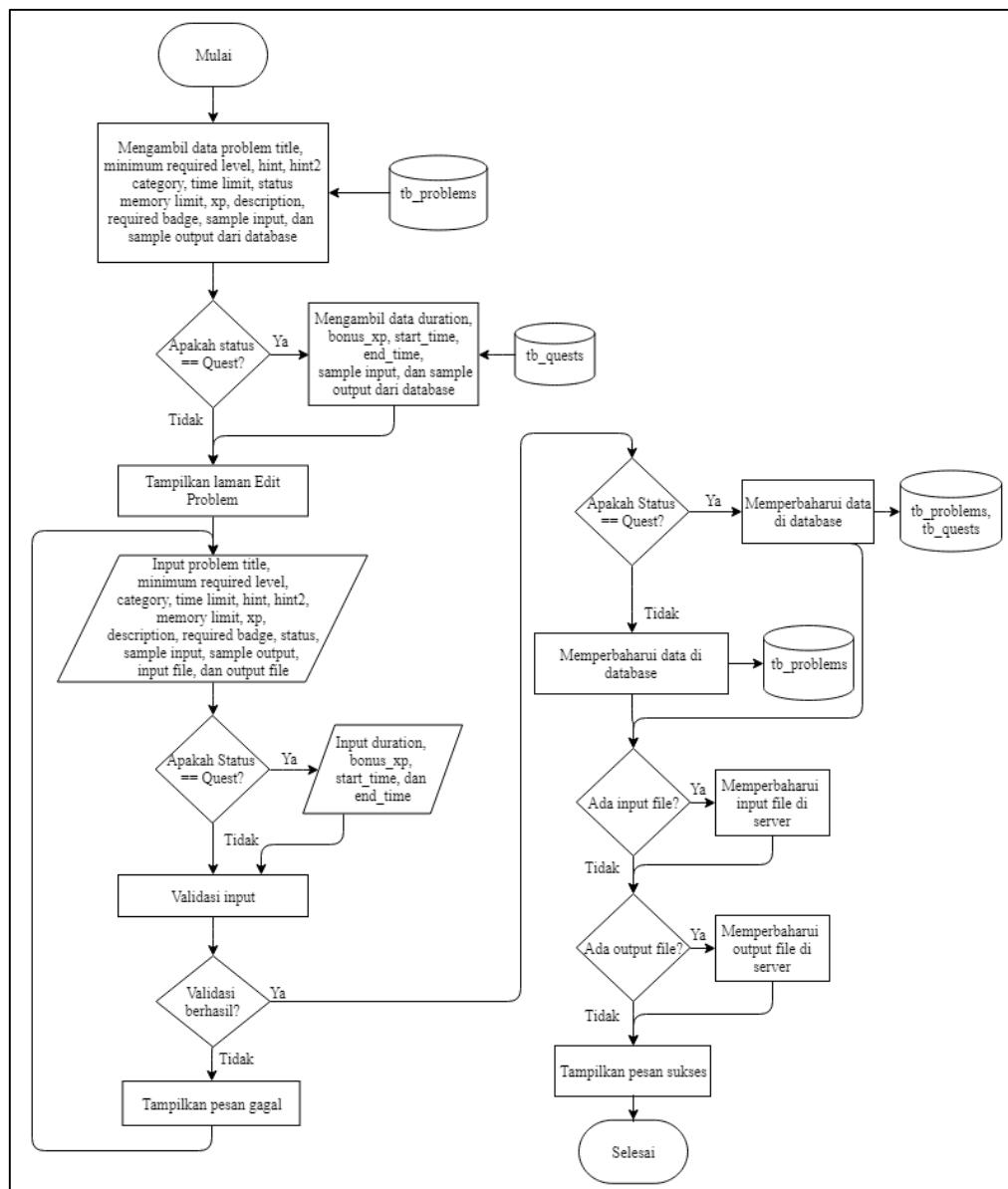
Gambar 3.24 Flowchart Manage Problem



Gambar 3.25 Flowchart Add Problem

Gambar 3.25 menunjukkan *flowchart* dari proses *add problem*. Pada proses ini admin dapat memasukkan data *problem title*, *minimum required level*, *category*, *time limit*, *xp*, *description*, *hint*, *hint2*, *status*, *required badge* jika ada, *sample input*, *sample output*, *input file*, dan *output file* pada *field*-nya masing-masing. Setelah itu, akan dilakukan pengecekan apakah *status* sama dengan *quest*. Jika tidak, maka dilanjutkan ke proses validasi input dst. Jika ya, maka admin akan memasukkan data tambahan berupa *duration*, *bonus xp*, *start time*, dan *end time*. Kemudian akan dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan meminta admin untuk memasukkan kembali data sesuai *field*-nya masing-masing. Jika validasi berhasil, maka akan dilakukan pengecekan apakah *status* sama dengan *quest*. Jika ya, maka data yang telah dimasukkan akan disimpan di tabel *problems* dan tabel *quests* dan dilanjutkandengan proses berikutnya. Jika tidak, maka data yang telah dimasukkan akan disimpan di tabel *problems*. Lalu *input file* dan *output file* akan disimpan di *server* dan menampilkan pesan sukses.

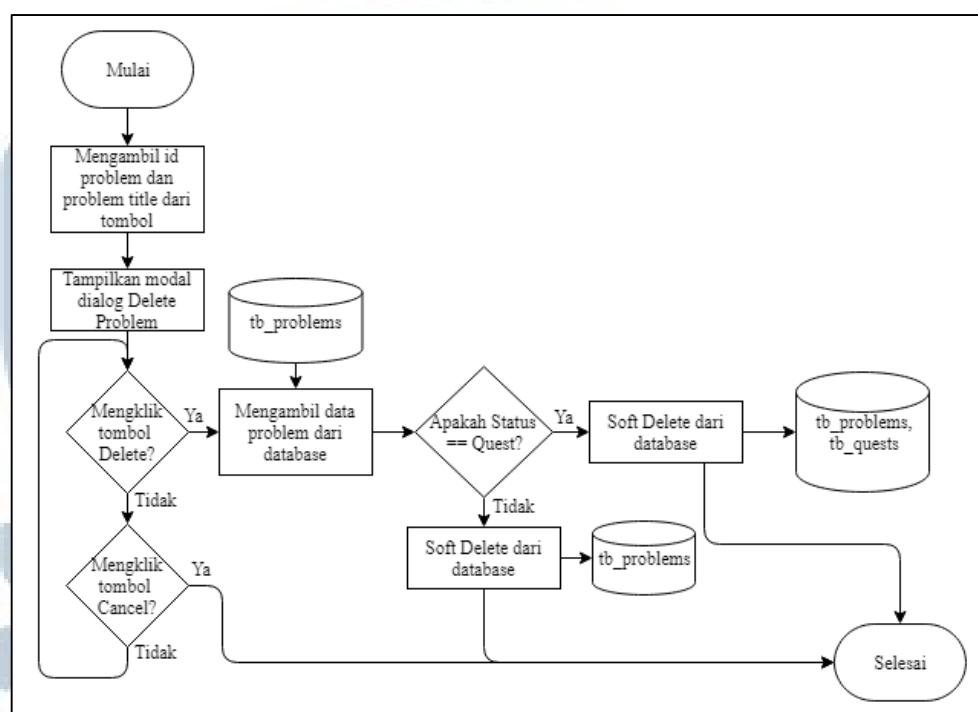




Gambar 3.26 Flowchart Edit Problem

Gambar 3.26 menunjukkan *flowchart* dari proses *edit problem*. Pada proses ini dilakukan pengambilan data *problem title*, *minimum required level*, *category*, *time limit*, *xp*, *status*, *hint*, *hint2*, *description*, *required badge* jika ada, *sample input*, *sample output*, *input file*, dan *output file*. Lalu dilakukan pengecekan apakah *status* sama dengan *quest*. Jika ya, maka dilakukan pengambilan data *duration*, *bonus_xp*, *start_time*, dan *end_time* kemudian akan ditampilkan di laman Edit Problem. Jika tidak, maka data sebelumnya yang telah diambil akan ditampilkan di laman Edit

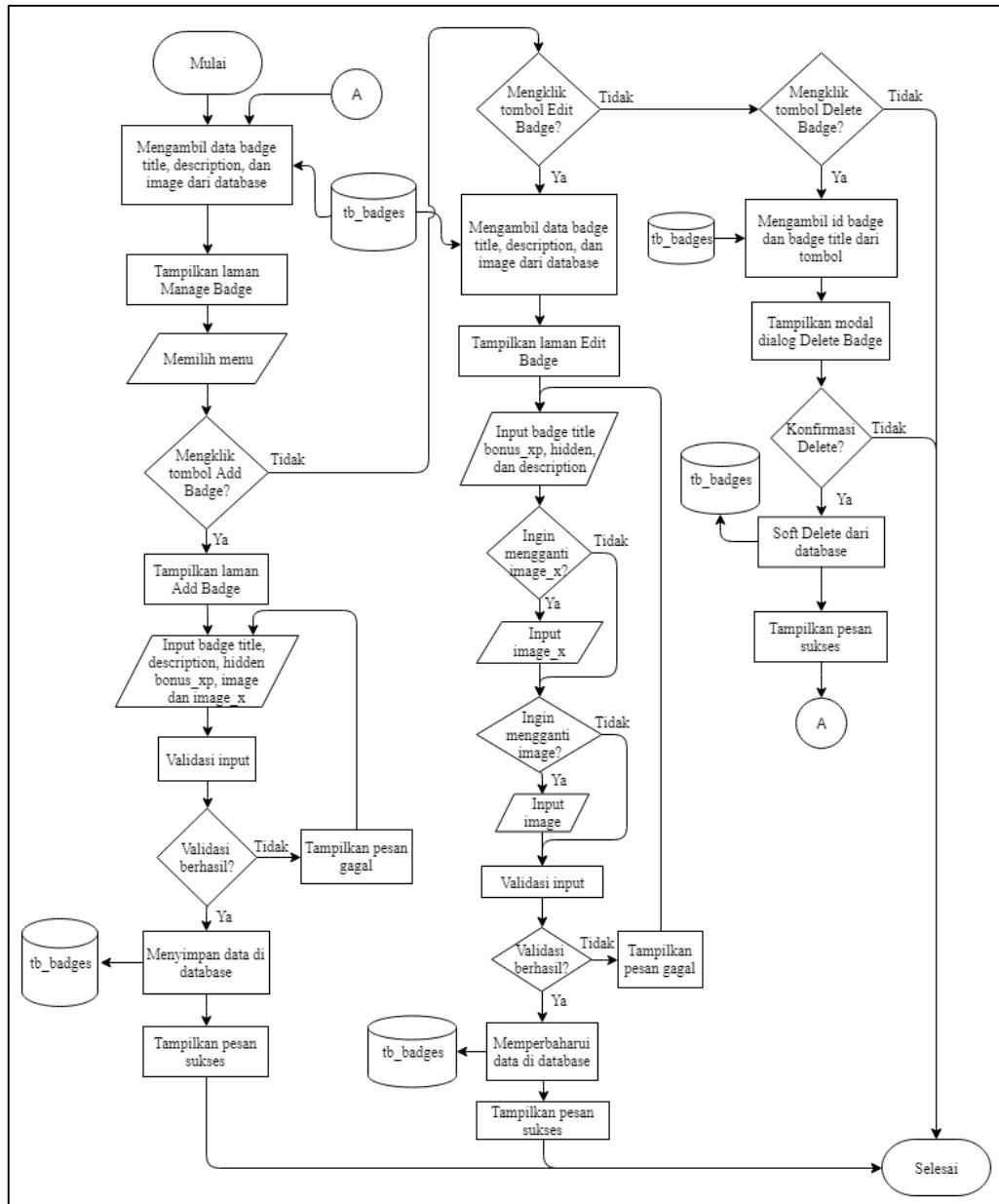
Problem. Setelah itu, admin dapat memasukkan data *problem title*, *minimum required level*, *category*, *time limit*, *xp*, *hint*, *hint2*, *description*, *required badge* jika ada, *sample input*, *sample output*, *input file*, dan *output file* baru pada *field*-nya masing-masing. Setelah itu, akan dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan meminta admin untuk memasukkan kembali data sesuai *field*-nya masing-masing. Jika validasi berhasil, maka akan dilakukan pengecekan apakah *status* sama dengan *quest*. Jika ya, maka data di tabel *problems* dan tabel *quests* akan diperbarui dengan data yang baru lalu dilanjutkan dengan proses berikutnya. Jika tidak, maka data di tabel *problems* akan diperbarui dengan data yang baru. Kemudian akan dilakukan pengecekan, apabila terdapat *input file*, maka data di server akan diperbarui. Apabila terdapat *output file*, maka data di server akan diperbarui dan menampilkan pesan sukses.



Gambar 3.27 Flowchart Delete Problem

Gambar 3.27 menunjukkan *flowchart* dari proses *delete problem*. Pada proses ini dilakukan pengambilan data *id problem* dan *problem title* dari tombol kemudian menampilkan *modal dialog*. Jika admin mengklik tombol Delete, maka akan dilakukan pengambilan data *problem* dari *database*. Lalu dilakukan pengecekan apakah *status* sama dengan *quest*. Jika ya, maka akan dilakukan *soft delete* pada data *problem* tersebut dari tabel *problems* dan tabel *quests* kemudian *modal dialog* akan tertutup. Jika tidak, maka akan dilakukan *soft delete* pada data *problem* tersebut dari tabel *problems* kemudian *modal dialog* akan tertutup. Jika admin mengklik tombol Cancel, maka *modal dialog* akan tertutup.



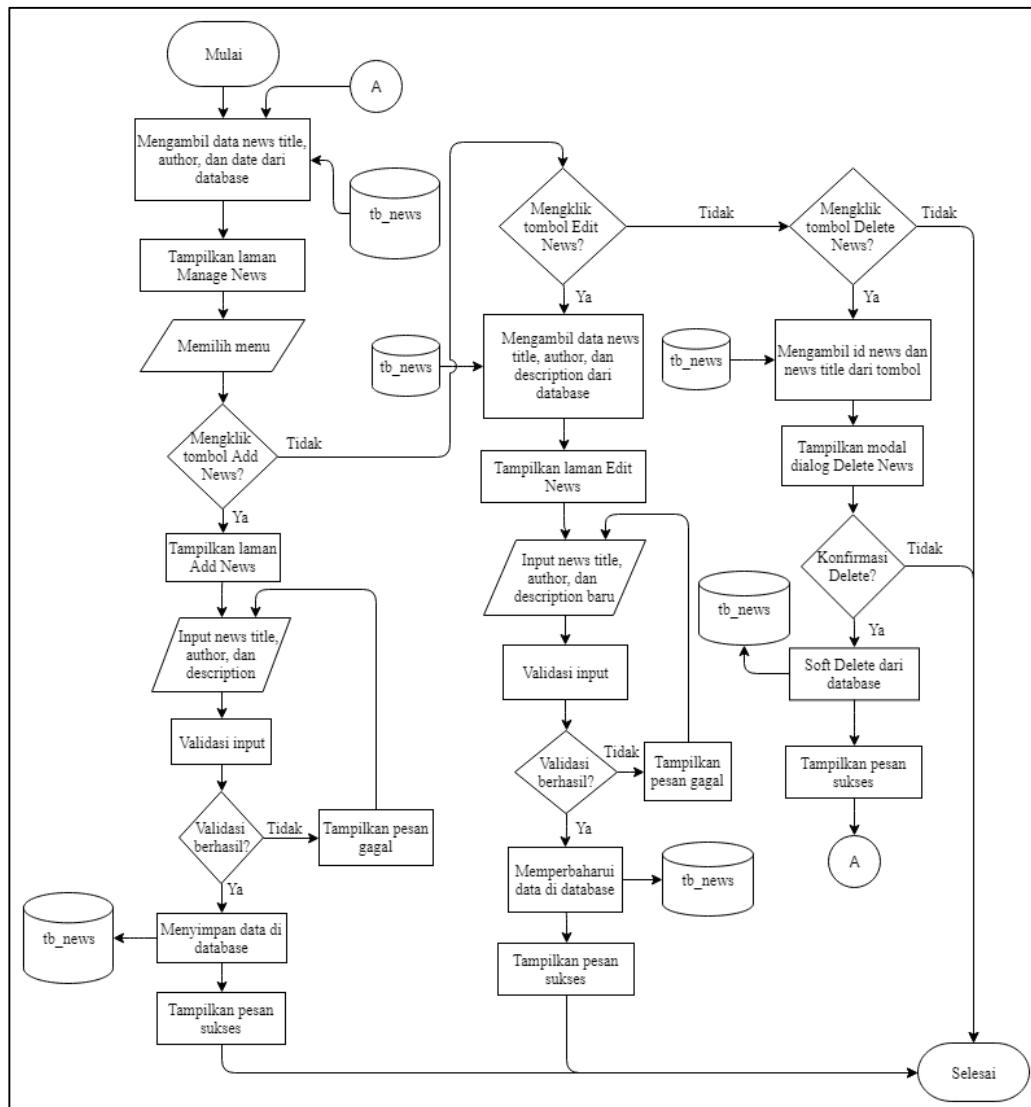


Gambar 3.28 Flowchart Manage Badge

Gambar 3.28 menunjukkan *flowchart* dari proses *manage badge*. Pada proses ini dilakukan pengambilan data *badge title*, *description*, dan *image* dari *database* kemudian ditampilkan ke admin. Setelah itu, admin memilih salah satu menu yang ada. Jika admin mengklik tombol *Add Badge*, maka akan ditampilkan laman *Add Badge* lalu admin dapat memasukkan data *badge title*, *description*, *bonus_xp*, *hidden*, *image_x* dan *image* pada *field* yang tersedia. Setelah itu, akan dilakukan

validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan admin akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data yang telah dimasukkan tadi akan disimpan di *database* dan akan ditampilkan pesan sukses. Jika admin mengklik tombol Edit Badge, maka akan dilakukan pengambilan data *badge title*, *description*, dan *image* dari *database* kemudian akan ditampilkan di laman Edit Badge. Admin dapat memasukkan data *badge title*, *bonus_xp*, *hidden*, dan *description* baru pada *field* yang tersedia. Jika admin ingin mengganti *image_x*, maka admin akan memilih *image* yang akan diunggah. Jika admin ingin mengganti *image*, maka admin akan memilih *image* yang akan diunggah. Setelah itu, dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan admin akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data di *database* akan diperbaharui dengan data yang baru dan akan ditampilkan pesan sukses. Jika admin mengklik tombol Delete Badge, maka akan dilakukan pengambilan *id badge* dan *badge title* dari *database* kemudian menampilkan *modal dialog*. Setelah itu, akan dilakukan konfirmasi untuk menghapus. Jika konfirmasinya ya, maka akan dilakukan *soft delete* pada data *badge* tersebut di *database* dan *modal dialog* akan tertutup. Jika konfirmasinya tidak, maka *modal dialog* akan tertutup.



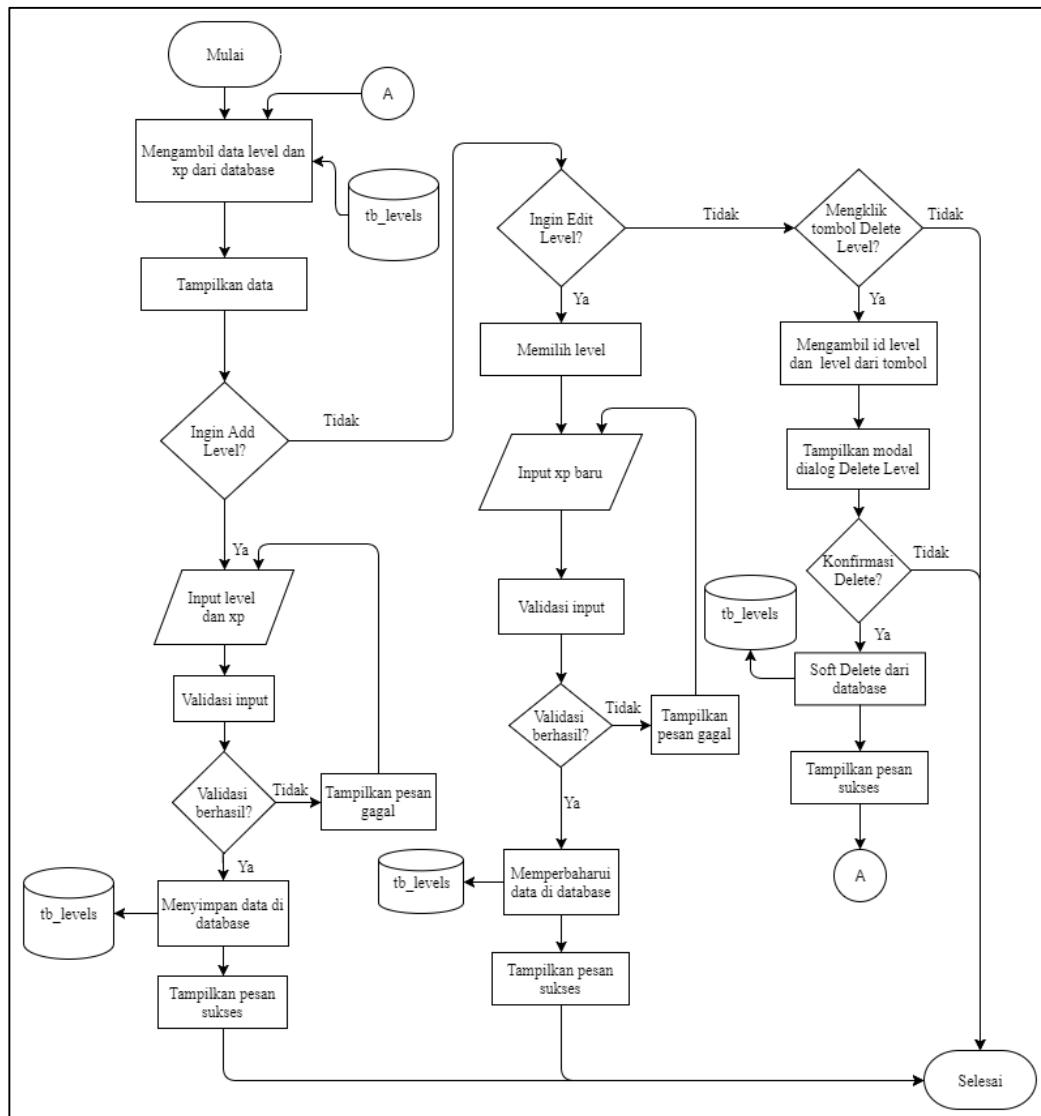


Gambar 3.29 Flowchart Manage News

Gambar 3.29 menunjukkan *flowchart* dari proses *manage news*. Pada proses ini dilakukan pengambilan data *news title*, *author*, dan *date* dari *database* kemudian ditampilkan di laman *Manage News*. Setelah itu, admin memilih salah satu menu yang ada. Jika admin mengklik tombol *Add News*, maka akan ditampilkan laman *Add News* lalu admin dapat memasukkan data *news title*, *author*, dan *description* pada *field* yang tersedia. Setelah itu, akan dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan admin akan diminta untuk memasukkan kembali data pada *field* yang

tersedia. Jika validasi berhasil, maka data yang telah dimasukkan tadi akan disimpan di *database* dan akan ditampilkan pesan sukses. Jika admin mengklik tombol Edit News, maka akan dilakukan pengambilan data *news title*, *author*, dan *description* dari *database* kemudian akan ditampilkan di laman Edit News. Admin dapat memasukkan data *news title*, *author*, dan *description* baru pada *field* yang tersedia. Setelah itu, akan dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan admin akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data di *database* akan diperbaharui dengan data yang baru dan akan ditampilkan pesan sukses. Jika admin mengklik tombol Delete News, maka akan dilakukan pengambilan *id news* dan *news title* dari *database* kemudian menampilkan *modal dialog*. Setelah itu, akan dilakukan konfirmasi untuk menghapus. Jika konfirmasinya ya, maka akan dilakukan *soft delete* pada data *news* tersebut di *database* dan *modal dialog* akan tertutup. Jika konfirmasinya tidak, maka *modal dialog* akan tertutup.

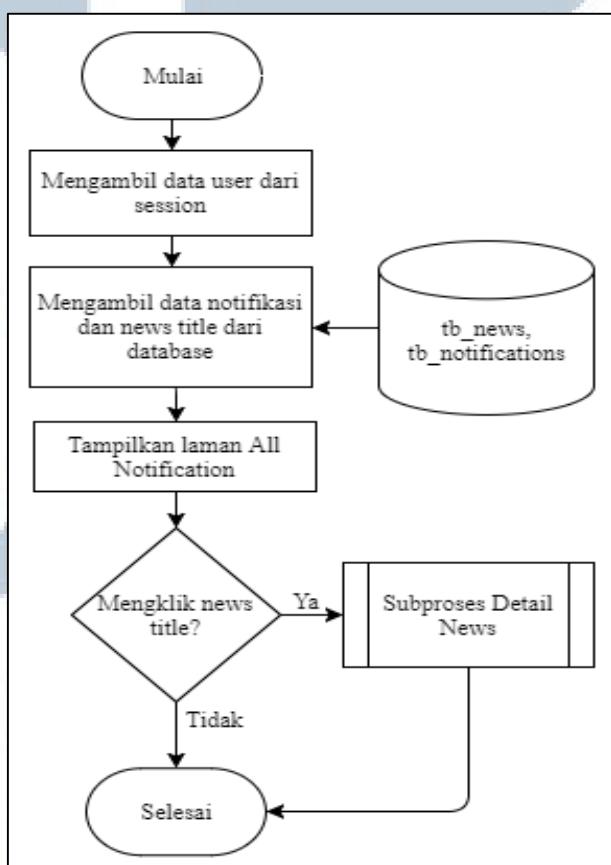




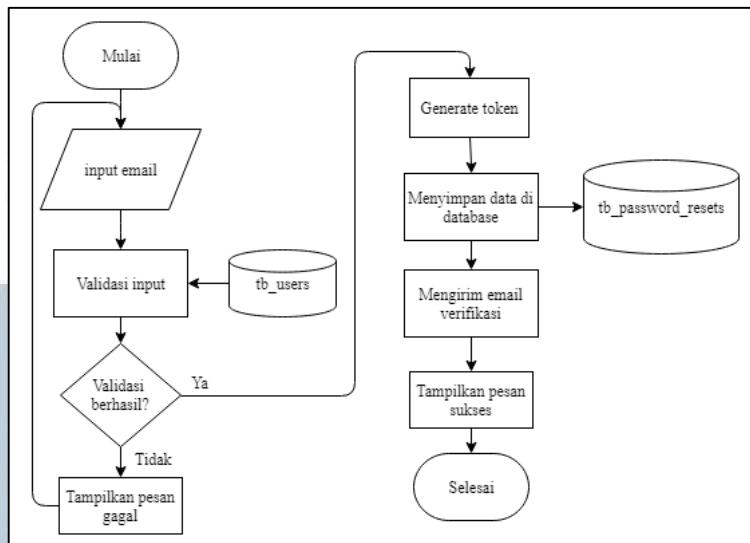
Gambar 3.30 Flowchart Manage Level

Gambar 3.30 menunjukkan *flowchart* dari proses *manage level*. Pada proses ini dilakukan pengambilan data *level*, dan *xp* dari *database* kemudian ditampilkan di laman *Manage Level*. Jika admin menambah *level*, maka admin dapat memasukkan data *level* dan *xp* pada *field* yang tersedia. Setelah itu, akan dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan admin akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data yang telah dimasukkan tadi akan disimpan di *database* dan akan ditampilkan pesan sukses.

Jika admin mengedit level, maka admin diminta untuk memilih data *level* yang akan diedit dan memasukkan *xp* baru pada *field* yang tersedia. Setelah itu, akan dilakukan validasi *input* untuk mengecek apakah semua *field* sudah terisi. Jika validasi gagal, maka akan ditampilkan pesan gagal dan admin akan diminta untuk memasukkan kembali data pada *field* yang tersedia. Jika validasi berhasil, maka data di *database* akan diperbarui dengan data yang baru dan akan ditampilkan pesan sukses. Jika admin menghapus level, maka akan dilakukan pengambilan *id level* dan *level* dari *database* kemudian menampilkan *modal dialog*. Setelah itu, akan dilakukan konfirmasi untuk menghapus. Jika konfirmasinya ya, maka akan dilakukan *soft delete* pada data *level* tersebut di *database* dan *modal dialog* akan tertutup. Jika konfirmasinya tidak, maka *modal dialog* akan tertutup.



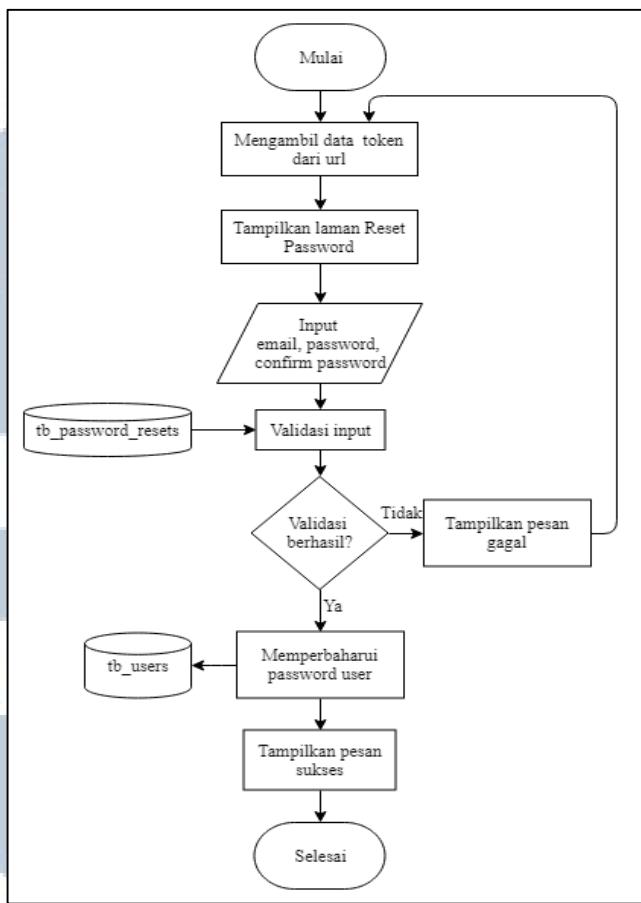
Gambar 3.31 Flowchart All Notification



Gambar 3.32 Flowchart Forgot Password

Gambar 3.32 menunjukkan *flowchart* dari proses *forgot password*. Pada proses ini, *user* menginput *email* kemudian akan dilakukan validasi *email* yang dimasukkan dengan *email* yang ada di *database*. Jika validasi berhasil, maka akan dilakukan *generate token* dan data yang telah dimasukkan beserta token akan disimpan di *database* lalu akan dikirimkan *email* untuk diverifikasi oleh *user* dan akan ditampilkan pesan sukses.

Gambar 3.33 menunjukkan *flowchart* dari proses *reset password*. Pada proses ini, dilakukan pengambilan data *token* dari *url* kemudian menampilkan laman Reset Password. *User* dapat menginput *email*, *password*, dan *confirm password* lalu dilakukan validasi terhadap *token* dan *email* yang di dengan data *token* dan *email* yang ada di tabel *password_resets* juga validasi kesamaan *password* dan *confirm password*. Jika validasi gagal, maka menampilkan pesan gagal dan proses kembali ke mengambil data *token* dari *url*. Jika validasi berhasil, maka akan memperbaharui *password user* yang baru di tabel *users* kemudian menampilkan pesan sukses.

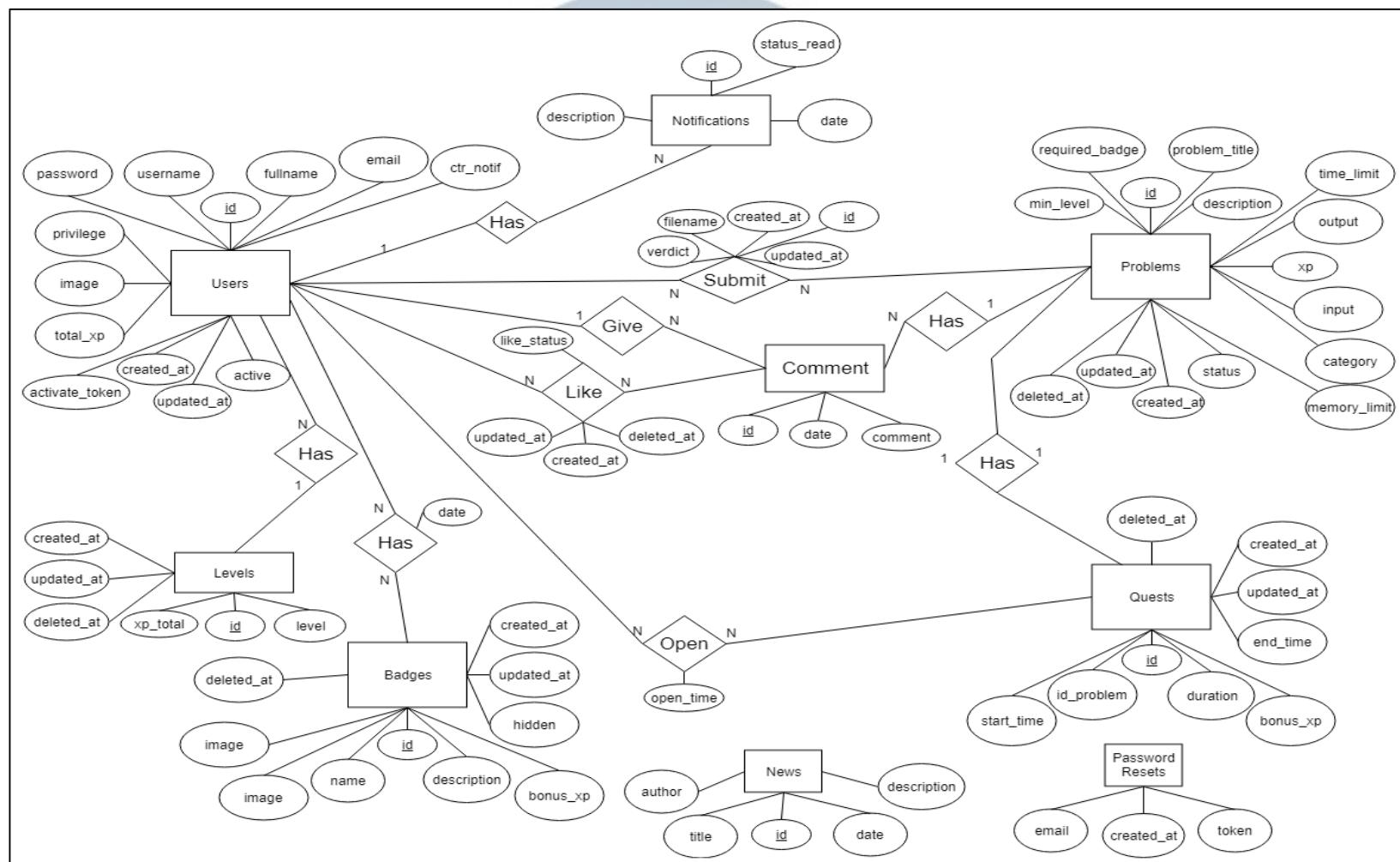


Gambar 3.33 Flowchart Reset Password

3.3.2 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram yang menunjukkan hubungan antar entitas. Gambar 3.34 menunjukkan *entity relationship diagram* dari *database* yang digunakan pada penelitian ini. Pada ERD ini, terdapat sembilan entitas yaitu entitas Users, Notifications, Problems, Badges, News, Levels, Comments, Password Resets dan Quests. Detil dari atribut-atribut dan hubungan dari tiap entitas dapat dilihat pada Gambar 3.34.

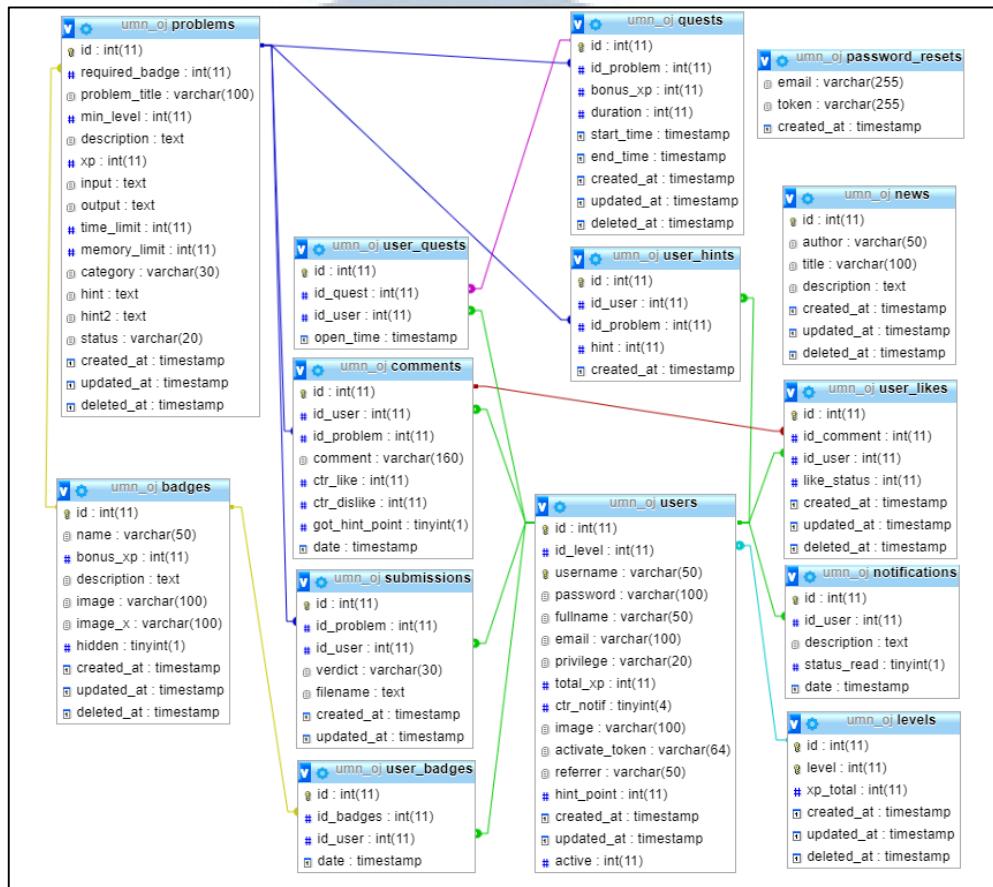
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.34 Entity Relationship Diagram Logical UMN OJ

MULTIMEDIA
NUSANTARA

3.3.3 Database Schema



Gambar 3.35 Database Schema UMN OJ

Gambar 3.35 menunjukkan *database schema* dari *database* yang digunakan pada penelitian ini. Untuk detil dari tabel-tabel yang digunakan akan dijelaskan pada subbab di bawah ini.

3.3.4 Struktur Tabel

Struktur tabel digunakan untuk menjelaskan data apa saja yang terdapat di dalam basis data untuk penelitian ini.

Tabel 3.1 Struktur Tabel Problems

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	required_badge	int	11	Ya	<i>Foreign Key</i> dari id Tabel Badges

Tabel 3.1 Struktur Tabel Problems (Lanjutan)

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
3.	problem_title	Varchar	100	Tidak	Nama dari soal
4.	min_level	int	11	Tidak	<i>Minimum required level</i> untuk masing-masing soal
5.	description	text		Tidak	Deskripsi dari soal
6.	xp	int	11	Tidak	Poin yang di- <i>assign</i> untuk masing-masing soal
7.	input	text	-	Tidak	Contoh <i>input</i> dari soal
8.	output	text	-	Tidak	Contoh output dari soal
9.	time_limit	int	11	Tidak	Waktu program yang ditentukan pada masing-masing soal
10.	memory_limit	int	11	Tidak	Memori program yang ditentukan pada masing-masing soal
11.	category	varchar	30	Tidak	Kategori dari soal
12.	status	varchar	20	Tidak	Status soal apakah soal biasa atau <i>quest</i>
13.	hint	text	-	Ya	<i>Highlight</i> dari <i>problem</i>
14.	Hint2	text	-	Ya	<i>Hint</i> berupa algoritma
15.	created_at	timestamp	-	Tidak	Waktu data pertama kali di- <i>insert</i>
16.	updated_at	timestamp	-	Tidak	Waktu data diubah
17.	deleted_at	timestamp	-	Ya	Waktu data dihapus

Tabel 3.1 merupakan tabel yang berfungsi untuk menyimpan data soal yang tersedia pada UMN OJ.

Tabel 3.2 merupakan tabel yang berfungsi untuk menyimpan data *submission* yang dilakukan oleh *user*.

Tabel 3.2 Struktur Tabel Submissions

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	id_problem	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Problems
3.	id_user	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Users
4.	verdict	varchar	20	Tidak	Status dari jawaban program yang diunggah
5.	filename	text	-	Tidak	Nama file dari program yang diunggah
6.	created_at	timestamp	-	Tidak	Waktu data pertama kali di- <i>insert</i>
7.	updated_at	timestamp	-	Tidak	Waktu data diubah

Tabel 3.3 Struktur Tabel Quests

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	id_problem	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Problems
3.	bonus_xp	int	11	Tidak	Bonus poin yang dapat diperoleh
4.	duration	int	20	Tidak	Durasi pengerjaan quest
5.	start_time	timestamp	-	Tidak	Waktu awal dimulainya quest
6.	end_time	timestamp	-	Tidak	Waktu selesai quest
7.	created_at	timestamp	-	Tidak	Waktu data pertama kali di- <i>insert</i>
8.	updated_at	timestamp	-	Tidak	Waktu data diubah
9.	deleted_at	timestamp	-	Ya	Waktu data dihapus

Tabel 3.3 merupakan tabel yang berfungsi untuk menyimpan data *quest* yang tersedia pada UMN OJ.

Tabel 3.4 Struktur Tabel Levels

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	level	int	11	Tidak	Daftar level yang ada
3.	xp_total	int	11	Tidak	Poin yang harus diperoleh di masing-masing level
4.	created_at	timestamp	-	Tidak	Waktu data pertama kali di- <i>insert</i>
5.	updated_at	timestamp	-	Tidak	Waktu data diubah
6.	deleted_at	timestamp	-	Ya	Waktu data dihapus

Tabel 3.4 merupakan tabel yang berfungsi untuk menyimpan data *level* yang tersedia pada UMN OJ.

Tabel 3.5 Struktur Tabel Badges

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	name	varchar	50	Tidak	Nama badges
3.	bonus_xp	int	11	Tidak	Bonus poin yang dapat diperoleh
4.	description	text	-	Tidak	Deskripsi badges
5.	image	varchar	100	Tidak	Path dari gambar badge
6.	image_x	varchar	100	Tidak	Path dari gambar badge abu-abu
7.	hidden	tinyint	1	Tidak	Penanda badge <i>hidden</i> atau tidak
8.	created_at	timestamp	-	Tidak	Waktu data pertama kali di- <i>insert</i>
9.	updated_at	timestamp	-	Tidak	Waktu data diubah
10.	deleted_at	timestamp	-	Ya	Waktu data dihapus

Tabel 3.5 merupakan tabel yang berfungsi untuk menyimpan data *badges* yang tersedia pada UMN OJ.

Tabel 3.6 Struktur Tabel Notifications

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	id_user	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Users
3.	description	text	11	Tidak	Deskripsi dari notifikasi
4.	status-read	tinyint	1	Tidak	Boolean untuk mengetahui status baca dari notifikasi
5.	date	timestamp	-	Tidak	Waktu mendapatkan notifikasi

Tabel 3.6 merupakan tabel yang berfungsi untuk menyimpan data notifikasi yang diterima oleh *user*.

Tabel 3.7 Struktur Tabel Comments

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	id_user	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Users
3.	id_problem	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Problems
4.	comment	varchar	160	Tidak	Komen dari <i>user</i>
5.	ctr_like	int	11	Tidak	Counter untuk menampung jumlah <i>like</i>
6.	ctr_dislike	int	11	Tidak	Counter untuk menampung jumlah <i>dislike</i>
7.	got_hint_point	tinyint	2	Tidak	Boolean untuk menandakan bahwa <i>user</i> sudah pernah mendapat <i>hint point</i>
8.	date	timestamp	-	Tidak	Waktu memberikan komen

Tabel 3.7 merupakan tabel yang berfungsi untuk menyimpan data komen yang dimasukan oleh *user* pada soal tertentu.

Tabel 3.8 Struktur Tabel News

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	Primary Key
2.	author	varchar	50	Ya	Nama pemberi news
3.	title	varchar	100	Tidak	Judul news
4.	description	text	-	Tidak	Deskripsi dari news
5.	created_at	timestamp	-	Tidak	Waktu posting news
6.	updated_at	timestamp	-	Tidak	Waktu data diubah
7.	deleted_at	timestamp	-	Ya	Waktu data dihapus

Tabel 3.8 merupakan tabel yang berfungsi untuk menyimpan data *news* yang di-post oleh admin.

Tabel 3.9 Struktur Tabel Users

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	Primary Key
2.	id_level	int	11	Tidak	Foreign Key dari id Tabel Levels
3.	username	int	11	Tidak	Username user
4.	password	int	11	Tidak	Kata sandi user
5.	fullname	int	11	Tidak	Nama lengkap user
6.	email	int	11	Tidak	Email user
7.	privilege	int	11	Tidak	Pembeda user dan admin
8.	total_xp	int	11	Tidak	Poin yang dimiliki user
9.	ctr_notif	tinyint	4	Tidak	Flag penanda notif user
10.	image	timestamp	-	Ya	Path dari foto user
11.	activate_token	varchar	64	Ya	Token untuk verifikasi email
12.	referrer	varchar	50	Ya	Username yang mengundang untuk bergabung di UMN OJ

Tabel 3.9 Struktur Tabel Users (Lanjutan)

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
13.	hint_point	int	11	Tidak	Jumlah poin yang dapat digunakan untuk membuka <i>hint</i> pada <i>problem</i>
14.	created_at	timestamp	-	Tidak	Waktu data pertama kali di- <i>insert</i>
15.	updated_at	timestamp	-	Tidak	Waktu data diubah
16.	active	int	11	Tidak	Status user yang sudah terverifikasi atau belum

Tabel 3.9 merupakan tabel yang berfungsi untuk menyimpan data *user* yang telah terdaftar di UMN OJ.

Tabel 3.10 Struktur Tabel User_Badges

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	id_badges	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Badges
3.	id_user	Int	11	Tidak	<i>Foreign Key</i> dari id Tabel Users
4.	date	timestamp	-	Tidak	Waktu mendapatkan badges

Tabel 3.10 merupakan tabel yang berfungsi untuk menyimpan data *badges* yang telah diperoleh oleh *user*.

Tabel 3.11 Struktur Tabel User_Quests

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	<i>Primary Key</i>
2.	id_quest	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Quests
3.	id_user	int	11	Tidak	<i>Foreign Key</i> dari id Tabel Users
4.	open_time	timestamp	-	Tidak	Waktu membuka quest

Tabel 3.11 merupakan tabel yang berfungsi untuk menyimpan data waktu *user* membuka *quest*.

Tabel 3.12 Struktur Tabel Password_Resets

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	email	varchar	255	Tidak	Email user
2.	token	varchar	255	Tidak	Token yang digunakan untuk reset
3.	created_at	timestamp	-	Ya	Waktu data pertama kali di-insert

Tabel 3.12 merupakan tabel yang berfungsi untuk menyimpan data *token* dan *email* untuk *user* mengubah *password*.

Tabel 3.13 Struktur Tabel User_Hints

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	Primary Key
2.	id_problem	int	11	Tidak	Foreign Key dari id Tabel Problems
3.	id_user	int	11	Tidak	Foreign Key dari id Tabel User
4.	hint	int	11	Tidak	Nomor kondisi yang akan membuka <i>hint</i> atau <i>hint2</i>
5.	created_at	timestamp	-	Tidak	Waktu data pertama kali di-insert

Tabel 3.13 merupakan tabel yang berfungsi untuk menyimpan data *hint* yang pernah dibuka oleh *user*.

Tabel 3.14 Struktur Tabel User_Likes

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
1.	id	int	11	Tidak	Primary Key
2.	id_comment	int	11	Tidak	Foreign Key dari id Tabel Comments
3.	id_user	int	11	Tidak	Foreign Key dari id Tabel User
4.	like_status	int	11	Tidak	Menandakan <i>like</i> atau <i>dislike</i>
5.	created_at	timestamp	-	Tidak	Waktu data pertama kali di-insert

Tabel 3.14 Struktur Tabel User_Likes (Lanjutan)

No.	Nama Kolom	Tipe Data	Panjang	Null	Keterangan
6.	updated_at	timestamp	-	Tidak	Waktu data diubah
7.	deleted_at	timestamp	-	Ya	Waktu data dihapus

Tabel 3.14 merupakan tabel yang berfungsi menyimpan data *user* yang memberi *like* atau *dislike* pada suatu komen.

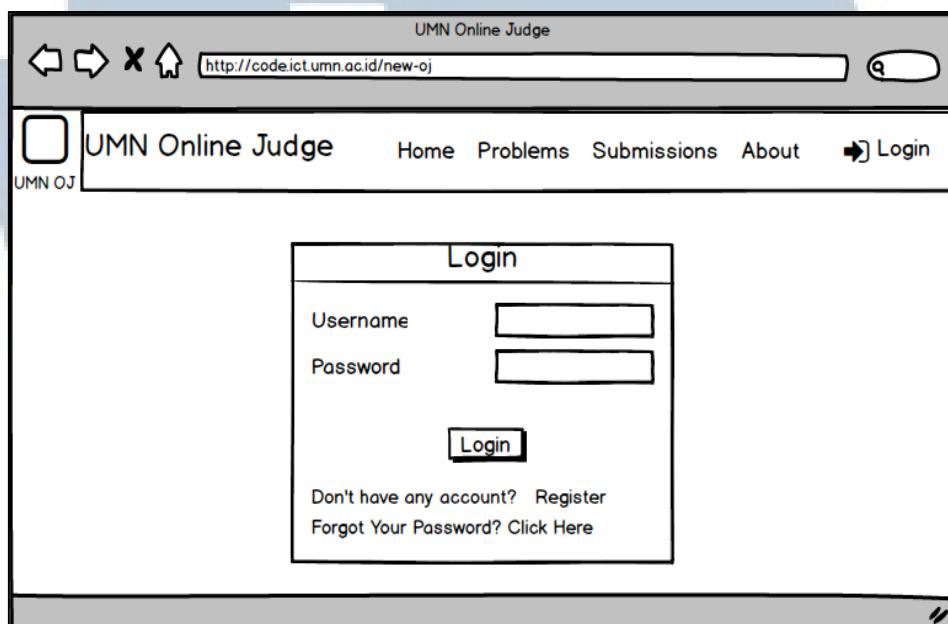
3.3.5 Design User Interface

Desain *user interface* dibuat untuk menggambarkan bagaimana tampilan aplikasi pada perancangan penelitian ini.

Gambar 3.36 Design User Interface Register

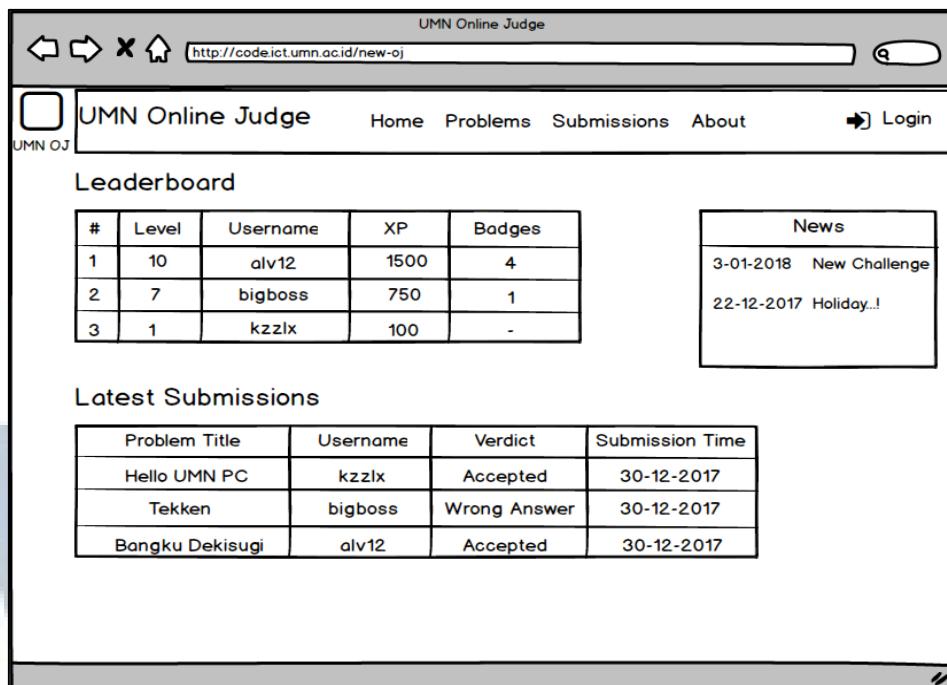
Gambar 3.36 menunjukkan *design user interface register*. Pada tampilan ini, terdapat *form register* yang berisi sepuluh *label* yaitu *label name*, *email*, *username*, *password*, *confirm password*, *referrer*, *forgot your password*, *already have an account*, dan *Login*. Selain itu, terdapat enam *field text box* yang dapat menerima masukan sesuai dengan *label* yang ada di sampingnya dan satu tombol *Register*. Jika tombol *Register* diklik, maka semua data yang telah dimasukkan ke masing-

masing *field* akan diproses untuk disimpan di *database*. Jika *user* telah memiliki akun, maka dapat mengklik *label Login* untuk masuk ke tampilan *login*. Jika *user* lupa *password* maka dapat melakukan *reset password* dengan mengklik *label click here* untuk masuk ke tampilan *reset password*. Untuk *field referrer* akan terisi secara otomatis apabila *user* tersebut merupakan *user* yang mendapat undangan.



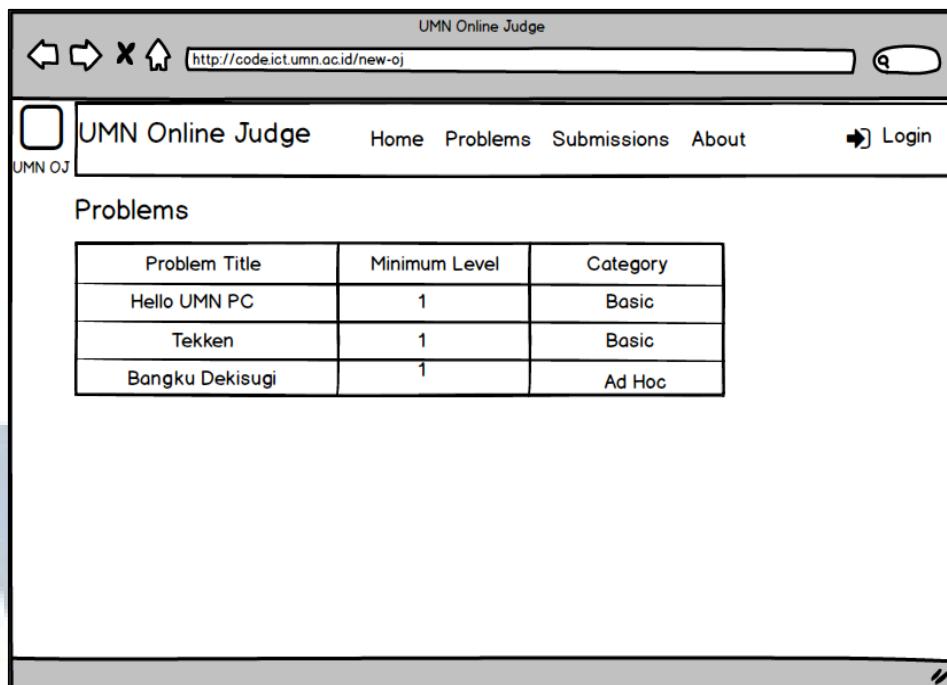
Gambar 3.37 Design User Interface Login

Gambar 3.37 menunjukkan *design user interface login*. Pada tampilan ini, terdapat *form login* yang berisi lima *label* yaitu *label username*, *password*, *forgot your password*, *don't have any account*, dan *Register*. Selain itu, terdapat juga dua *field text box* yang dapat menerima masukan sesuai dengan *label* yang ada di sampingnya dan satu tombol *Login*. Jika tombol *Login* diklik, maka akan memproses data yang telah dimasukkan pada *field* yang ada. Jika *user* belum memiliki akun, maka dapat mengklik *label Register* untuk masuk ke tampilan *register*. Jika *user* lupa *password* maka dapat melakukan *reset password* dengan mengklik *label click here* untuk masuk ke tampilan *reset password*.



Gambar 3.38 Design User Interface Home Guest

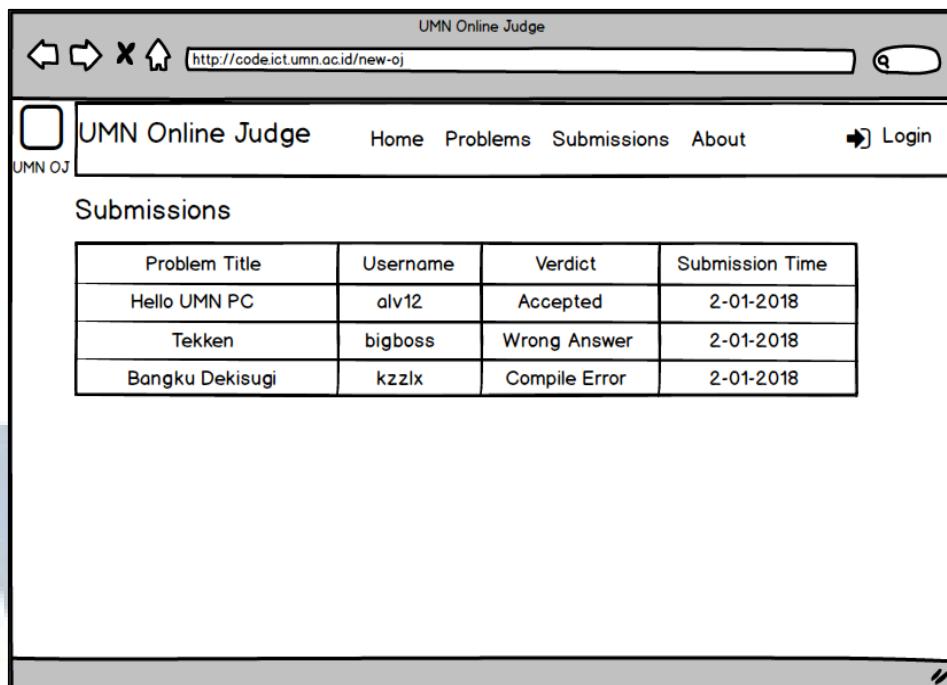
Gambar 3.38 menunjukkan *design user interface home guest*. Pada tampilan ini, terdapat tiga *label* yaitu *label leaderboard*, *latest submissions*, *news*, dua tabel yaitu tabel *Leaderboard* yang berisi data *level*, *username*, *xp*, dan *badges*. Serta tabel *Latest Submissions* yang berisi data *problem title*, *username*, *verdict*, dan *submission time*. Terdapat pula sebuah kotak berisi *news title* dan waktu *postingnya* yang jika diklik, maka akan masuk ke tampilan *detail news*. Selain itu, terdapat juga tab menu seperti menu *Problem* dan *Submission*, juga icon menu *Login*. Jika tab menu *Problem* diklik, maka akan masuk ke tampilan *Problems*. Jika tab menu *Submissions* diklik, maka akan masuk ke tampilan *Submissions*. Jika tab menu *About* diklik, maka akan masuk ke tampilan laman *About*. Jika icon menu *Login* diklik, maka akan masuk ke tampilan *login*.



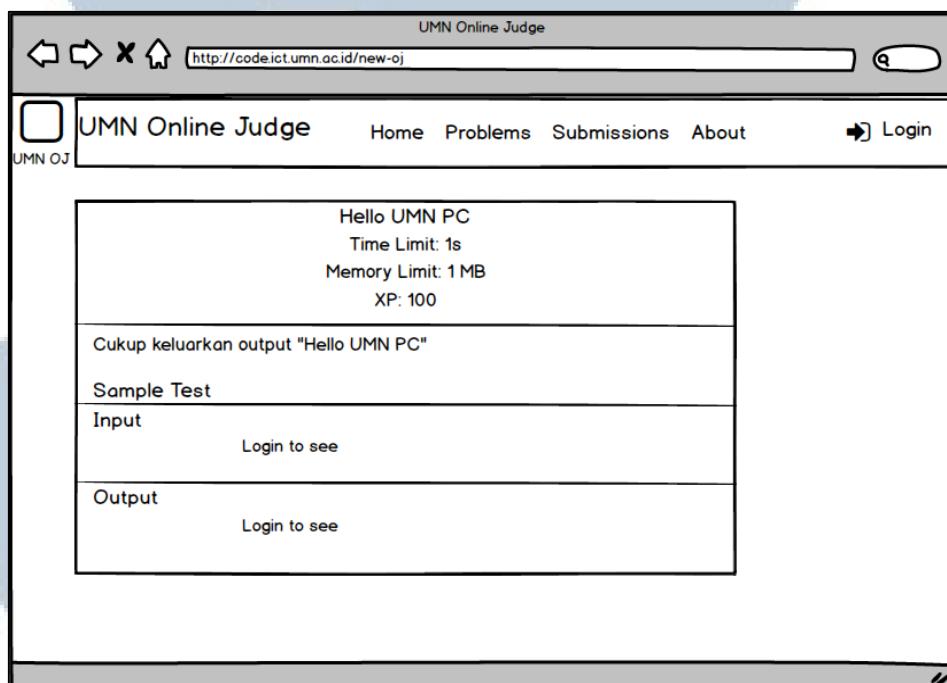
Gambar 3.39 Design User Interface Problems Guest

Gambar 3.39 menunjukkan *design user interface problem guest*. Pada tampilan ini, terdapat satu *label* yaitu *label Problems* dan satu tabel yang berisi data *problem title*, *minimum level*, dan *category*. Jika salah satu *problem title* diklik, maka akan masuk ke tampilan *detail problem guest* atau *detail quest guest* bergantung pada jenis *problem* yang diklik.

Gambar 3.40 menunjukkan *design user interface submissions guest*. Pada tampilan ini, terdapat satu *label* yaitu *label submission* dan satu tabel yang berisi data *problem title*, *username*, *verdict*, dan *submission time*. Jika salah satu *problem title* diklik, maka akan masuk ke tampilan *detail problem guest* atau *detail quest guest* bergantung pada jenis *problem* yang diklik. Jika salah satu *username* diklik, maka akan masuk ke tampilan *profile user* yang *username*-nya diklik.

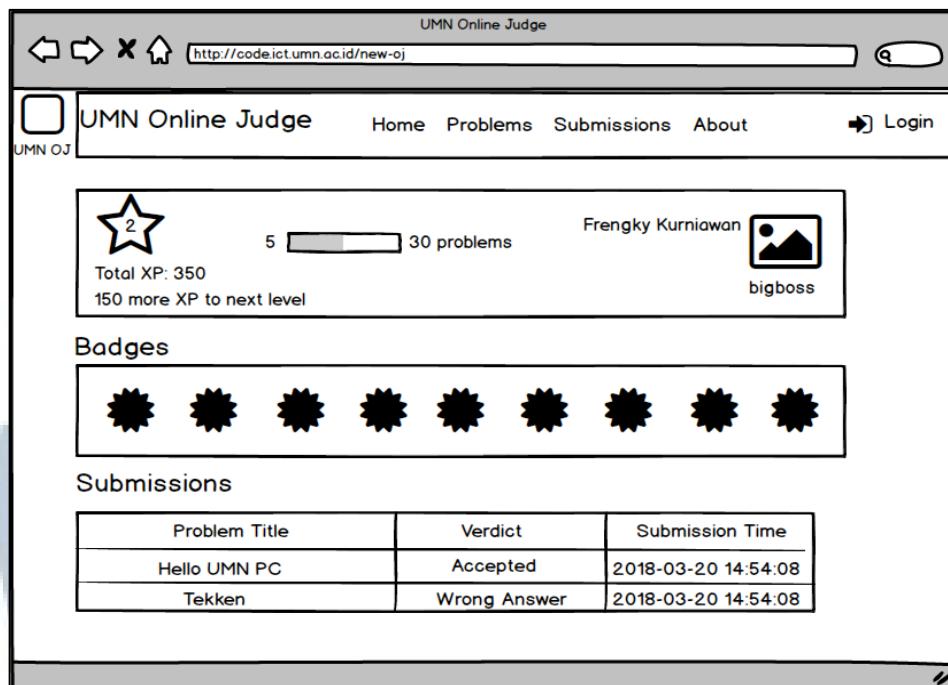


Gambar 3.40 Design User Interface Submissions Guest



Gambar 3.41 Design User Interface Detail Problem Guest

Gambar 3.41 menunjukkan *design user interface detail problem guest*. Pada tampilan ini, terdapat tujuh *label* yaitu *label problem title*, *time limit*, *memory limit*, *xp*, *sample test*, *input*, dan *output*. Selain itu, terdapat juga dua *field text area*.

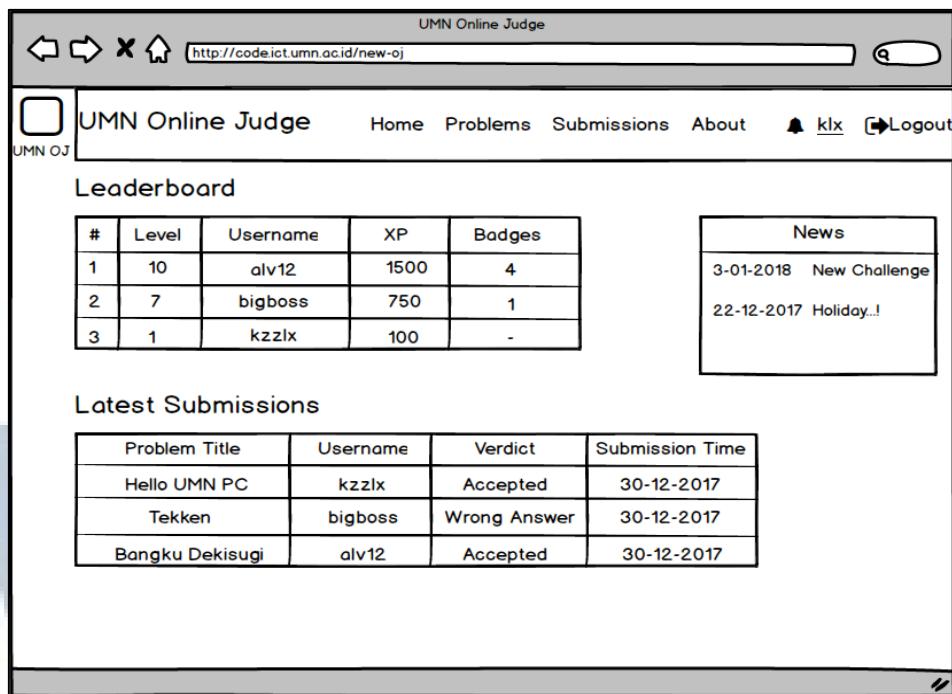


Problem Title	Verdict	Submission Time
Hello UMN PC	Accepted	2018-03-20 14:54:08
Tekken	Wrong Answer	2018-03-20 14:54:08

Gambar 3.42 Design User Interface Profile User

Gambar 3.42 menunjukkan *design user interface profile user*. Pada tampilan ini, terdapat satu *profile image*, tujuh *label* yaitu *label username*, nama, *total xp*, *total xp* yang dibutuhkan untuk naik *level* berikutnya, jumlah *problem* yang telah dikerjakan dari *total problem*, *badges*, dan *submissions*. Terdapat juga satu *progress bar*, satu *icon level*, satu kotak yang berisi *list badges* dan satu tabel yang berisi data *problem title*, *verdict*, dan *submission time*. Jika salah satu *problem title* diklik, maka akan masuk ke tampilan *detail problem guest* atau *detail quest guest* bergantung pada jenis *problem* yang diklik.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.43 Design User Interface Home User

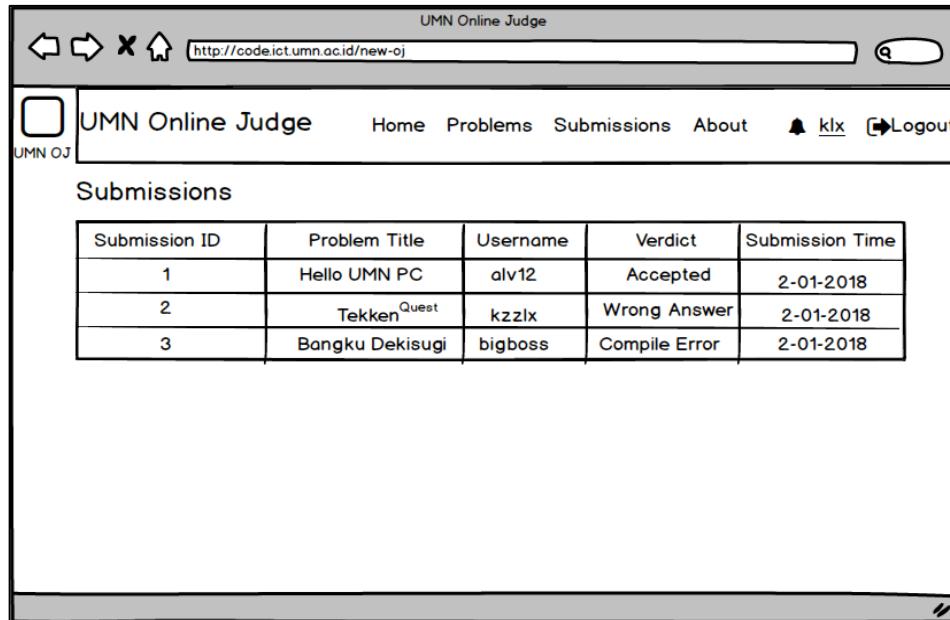
Gambar 3.43 menunjukkan *design user interface home user*. Pada tampilan ini, terdapat empat *label* yaitu *label leaderboard*, *latest submissions*, *news*, *username*, dua tabel yaitu tabel *Leaderboard* yang berisi data *level*, *username*, *xp*, dan *badges*. Serta tabel *Latest Submissions* yang berisi data *problem title*, *username*, *verdict*, dan *submission time*. Terdapat pula sebuah kotak berisi *news title* dan waktu *posting*-nya yang jika diklik, akan masuk ke tampilan *detail news*. Selain itu, terdapat juga tab menu seperti menu *Problem* dan *Submission*, juga *icon* menu *Logout* dan *Notification*. Jika tab menu *Problem* diklik, maka akan masuk ke tampilan *Problems*. Jika tab menu *Submission* diklik, maka akan masuk ke tampilan *Submissions*. Jika tab menu *About* diklik, maka akan masuk ke tampilan *About*. Jika *icon* menu *Logout* diklik, maka akan masuk ke tampilan *login*. Jika *icon* menu *Notification* diklik, maka akan menampilkan kotak berisi *list notification*. Jika *label username* diklik, maka akan masuk ke tampilan *Profile*.

Problem Title	Minimum Level	Category	Verdict
Hello UMN PC	1	Basic	Not yet
Tekken Quest	-	-	Wrong Answer
Bangku Dekisugi	1	Basic	Accepted

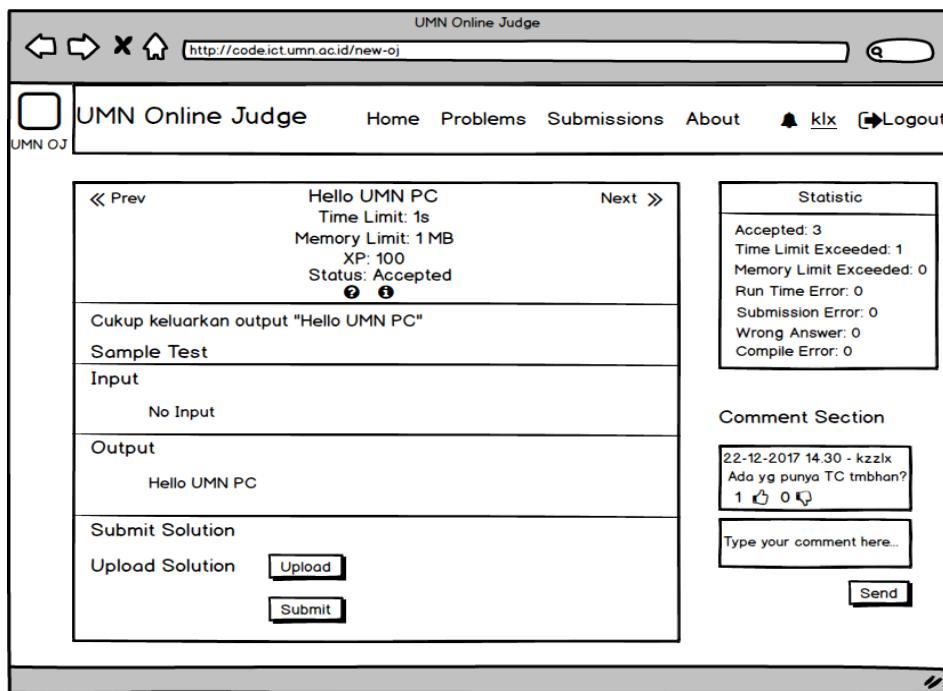
Gambar 3.44 Design User Interface Problems

Gambar 3.44 menunjukkan *design user interface problems*. Pada tampilan ini, terdapat satu *label* yaitu *label problems* dan satu tabel yang berisi data *problem title*, *minimum level*, *category*, dan *verdict*. Jika salah satu *problem title* diklik, maka akan masuk ke tampilan *detail problem guest* atau *detail quest guest* bergantung pada jenis *problem* yang diklik. *Problem* yang merupakan *quest* ditandai dengan tulisan *Quest* di dekat *problem title* tersebut.

Gambar 3.45 menunjukkan *design user interface submissions*. Pada tampilan ini, terdapat satu *label* yaitu *label submission* dan satu tabel yang berisi data *submission id*, *problem title*, *username*, *verdict*, dan *submission time*. Jika salah satu *problem title* diklik, maka akan masuk ke tampilan *detail problem guest* atau *detail quest guest* bergantung pada jenis *problem* yang diklik. Jika salah satu *username* diklik, maka akan masuk ke tampilan *profile user* yang *username*-nya diklik. *Problem* yang merupakan *quest* ditandai dengan tulisan *Quest* di dekat *problem title* tersebut. *Submission id* dapat diklik apabila *submission* tersebut merupakan *submission* dari *user* yang sedang *login*.



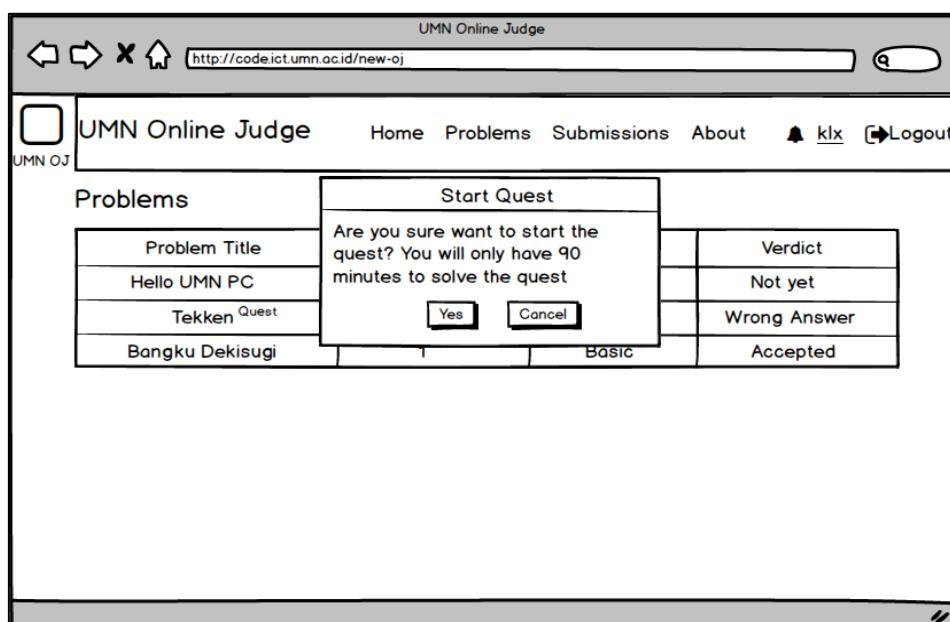
Gambar 3.45 Design User Interface Submissions



Gambar 3.46 Design User Interface Detail Problem

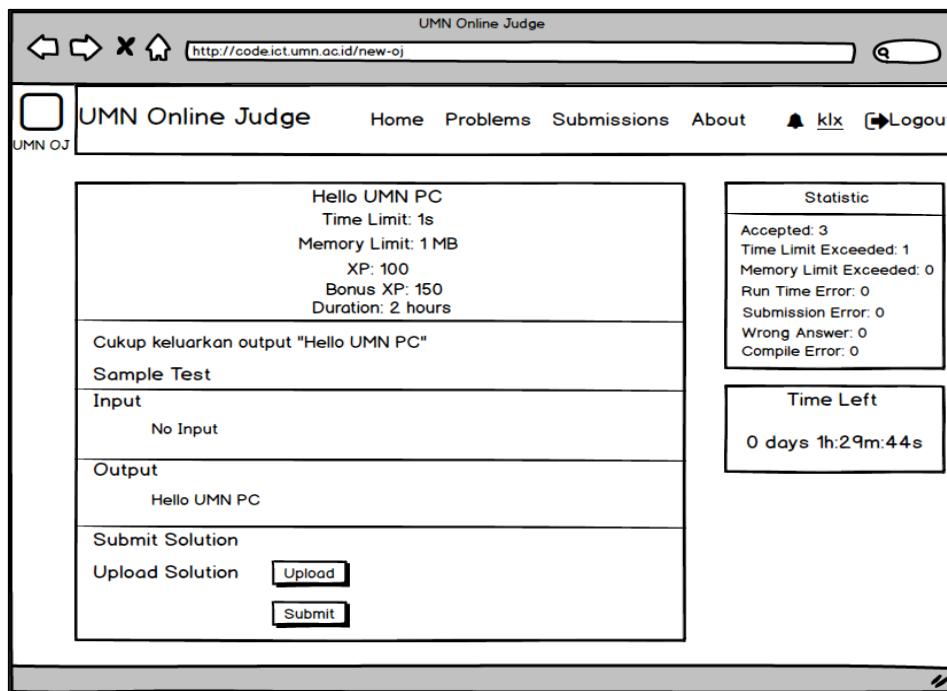
Gambar 3.46 menunjukkan *design user interface detail problem*. Pada tampilan ini, terdapat dua belas *label* yaitu *label problem title*, *time limit*, *memory limit*, *xp*, *status*, *sample test*, *input*, *output*, *submit solution*, *upload solution*, *statistic*, dan *comment section*. Terdapat juga ikon tanda tanya dan tanda seru. Jika ikon tanda tanya atau tanda seru diklik, maka akan menampilkan *hint* atau *highlight*.

dari *problem* tersebut. Selain itu, terdapat juga dua *field text area* untuk *input* dan *output*, empat tombol yaitu tombol Next, Prev, Upload dan Submit. Jika tombol Next diklik, maka akan membuka *detail* dari *problem* selanjutnya. Jika tombol Prev diklik, maka akan membuka *detail* dari *problem* sebelumnya. Jika tombol *upload* diklik, maka akan memilih *file* jawaban. Jika tombol Submit diklik, maka *file* jawaban yang telah dipilih akan diunggah ke *server*. Terdapat pula sebuah kotak berisi daftar *verdict* dari jawaban yang telah di-submit untuk *problem* yang sedang dibuka. Dan terdapat *field text box* untuk mengisi komentar dan satu tombol Send, yang jika diklik akan menyimpan komentar yang telah dimasukkan ke *database*. Serta terdapat *field text box* untuk membaca komentar yang masuk pada *problem* yang sedang dibuka.



Gambar 3.47 Design User Interface Modal Dialog Quest Confirmation

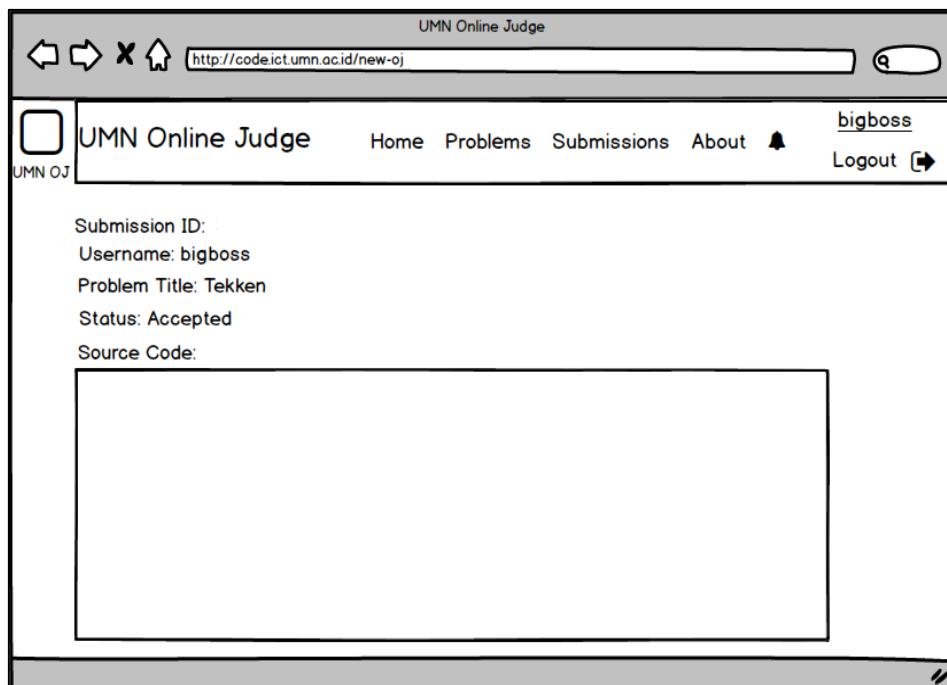
Gambar 3.47 menunjukkan *design user interface modal dialog quest confirmation*. Pada *modal dialog* ini, *user* akan diminta konfirmasi apakah *user* yakin untuk membuka *quest* atau tidak. Jika *user* mengklik *button Yes*, maka *user* akan masuk ke laman Detail Quest.



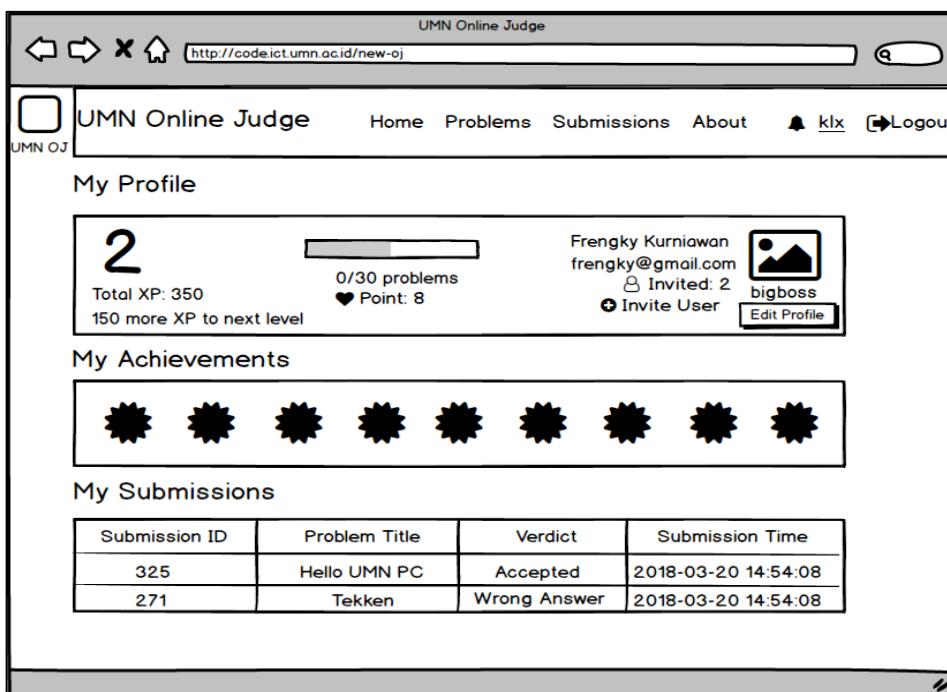
Gambar 3.48 Design User Interface Detail Quest

Gambar 3.48 menunjukkan *design user interface detail quest*. Pada tampilan ini, terdapat dua belas *label* yaitu *label problem title*, *time limit*, *memory limit*, *xp*, *bonus xp*, *duration*, *sample test*, *input*, *output*, *submit solution*, *upload solution*, dan *statistic*. Selain itu, terdapat juga dua *field text area* untuk *input* dan *output*, dua tombol yaitu tombol *Upload* dan *Submit*. Jika tombol *Upload* diklik, maka akan memilih *file* jawaban. Jika tombol *Submit* diklik, maka *file* jawaban yang telah dipilih akan diunggah ke *server*. Terdapat pula dua kotak berisi daftar *verdict* dari jawaban yang telah di-*submit* untuk *problem* yang sedang dibuka dan waktu yang tersisa untuk melakukan *submit* jawaban dari *quest* yang sedang dibuka.

Gambar 3.49 menunjukkan *design user interface detail submission*. Pada tampilan ini, terdapat lima *label* yaitu *label submission id*, *username*, *problem title*, *status*, dan *source code*. Dan terdapat sebuah kotak yang berisi *source code* jawaban untuk *submission id* tersebut.



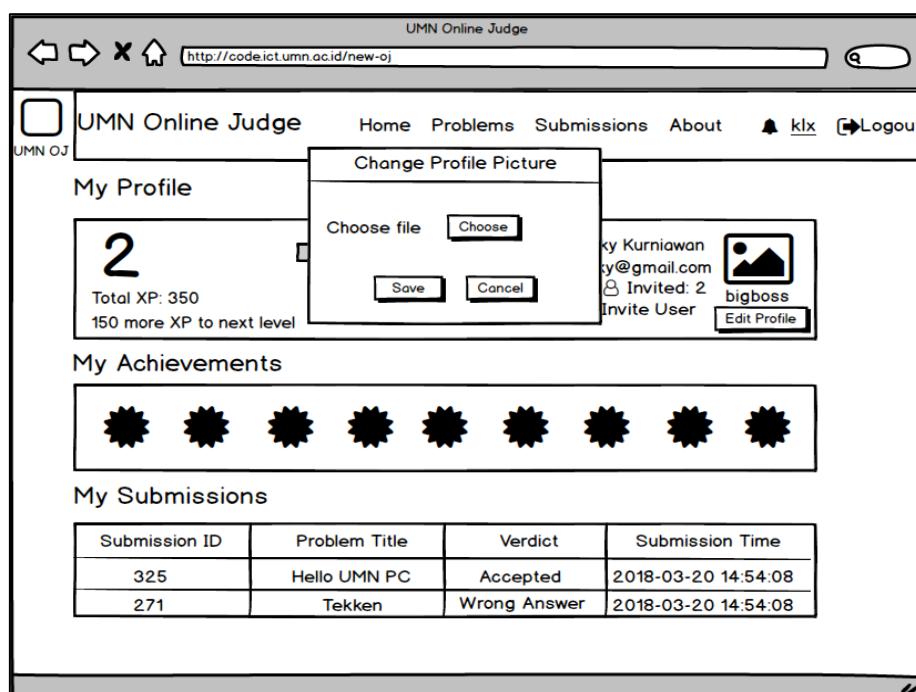
Gambar 3.49 Design User Interface Detail Submission



Gambar 3.50 Design User Interface Profile

Gambar 3.50 menunjukkan *design user interface profile*. Pada tampilan ini, terdapat satu *profile image*, tiga belas *label* yaitu *label level*, *username*, nama, *email*, *total xp*, *point*, *invited*, *total xp* yang dibutuhkan untuk naik ke *level* berikutnya, jumlah *problem* yang telah dikerjakan dari *total problem*, *My Profile*,

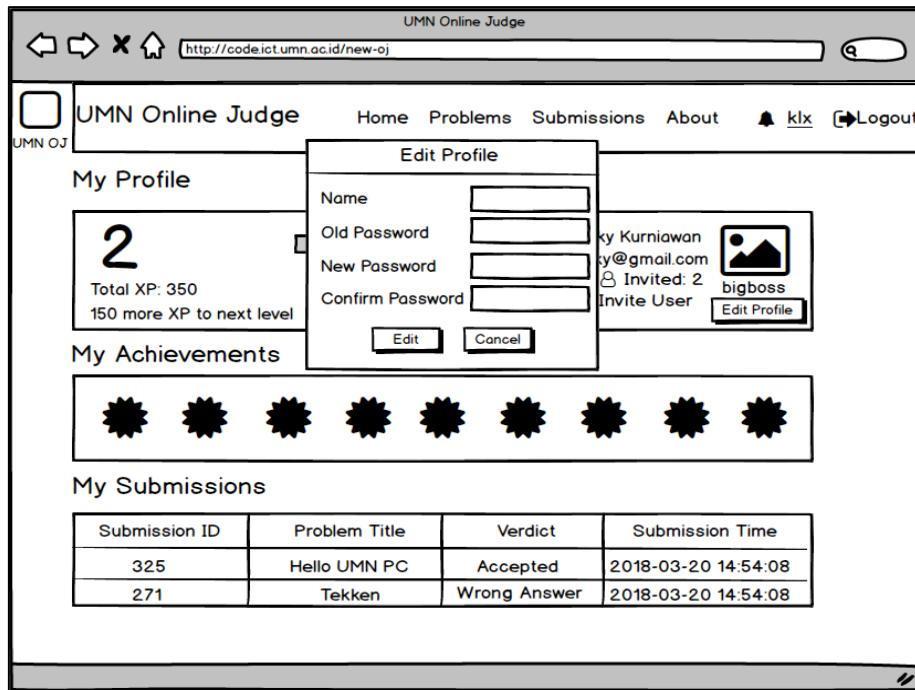
My Achievements, *My Submissions*, dan *total xp* yang diperlukan untuk naik ke *level* berikutnya. Terdapat juga satu *progress bar*, satu kotak yang berisi *list badges* dan satu tabel yang berisi data *submission id*, *problem title*, *verdict* dan *submission time*. Selain itu, terdapat dua yaitu tombol *Edit Profile* yang jika diklik, maka akan menampilkan tampilan *modal dialog* *Edit Profile* dan tombol *Invite User* yang jika diklik, maka akan menampilkan tampilan *modal dialog* *Invite User*. Jika *profile image* diklik, maka akan menampilkan tampilan *modal dialog* *Change Profile Picture*. Jika salah satu *submission id* diklik, maka akan masuk ke tampilan *detail submission*. Jika salah satu *problem title* diklik, maka akan masuk ke tampilan *detail problem* atau *detail quest* bergantung pada jenis *problem* yang diklik.



Gambar 3.51 Design User Interface Change Picture

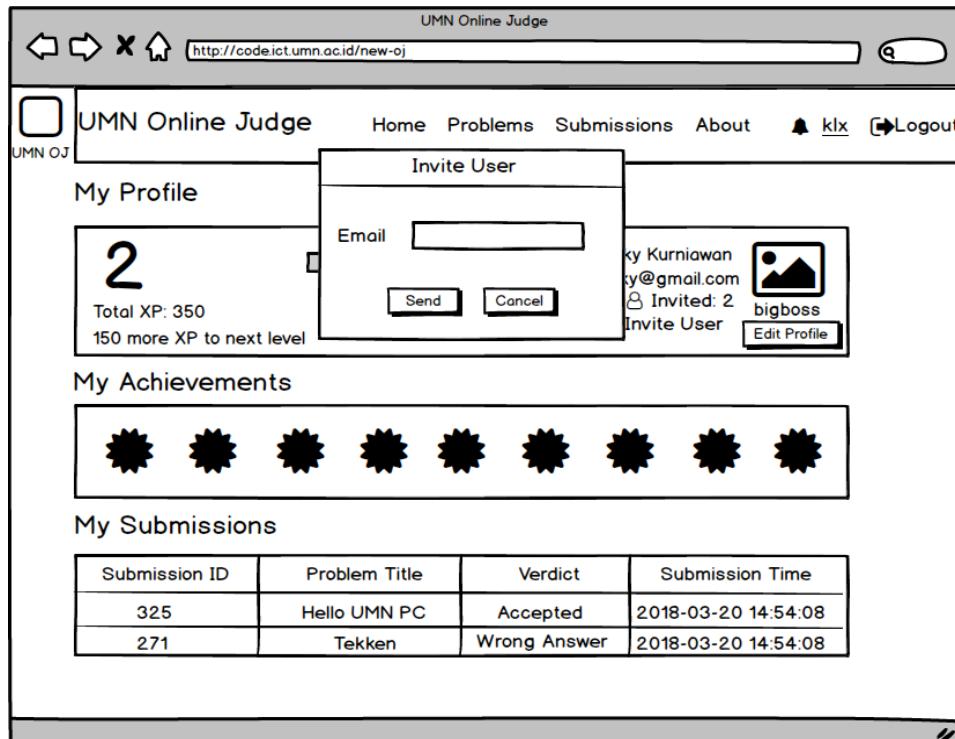
Gambar 3.51 menunjukkan *design user interface change picture*. Tampilan ini merupakan tampilan dari fungsi klik *profile image* pada tampilan *profile*. Tampilan ini merupakan *modal dialog* yang berisi tombol untuk memilih *file* gambar baru yang akan dijadikan *profile image* serta tombol *Save* dan *Cancel*. Jika

tombol Save diklik, maka data gambar baru akan diperbarui di *database*. Jika tombol Cancel diklik, maka *modal dialog* Change Profile Picture akan tertutup.



Gambar 3.52 Design User Interface Edit Profile

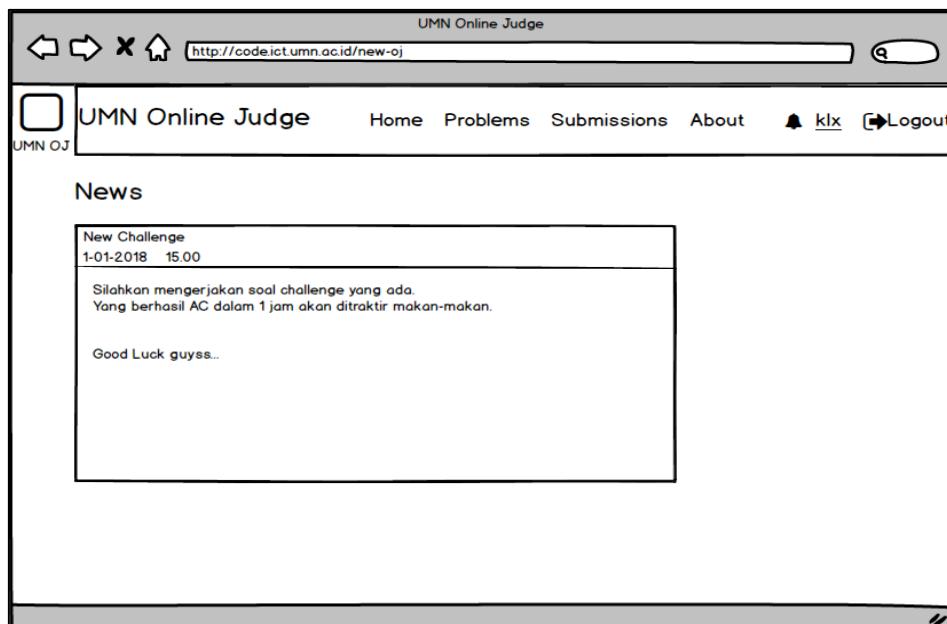
Gambar 3.52 menunjukkan *design user interface edit profile*. Tampilan ini merupakan tampilan dari fungsi klik *edit profile* pada tampilan *profile*. Tampilan ini merupakan *modal dialog* yang berisi empat *label* yaitu *label name*, *old password*, *new password*, dan *confirm password*. Terdapat juga empat *field text box* yang dapat menerima masukan data sesuai dengan *label* yang ada di sampingnya serta tombol *Edit* dan *Cancel*. Jika tombol *Edit* diklik, maka data yang baru dimasukkan akan diperbarui di *database*. Jika tombol *Cancel* diklik, maka *modal dialog* *Edit Profile* akan tertutup.



Gambar 3.53 Design User Interface Invite User

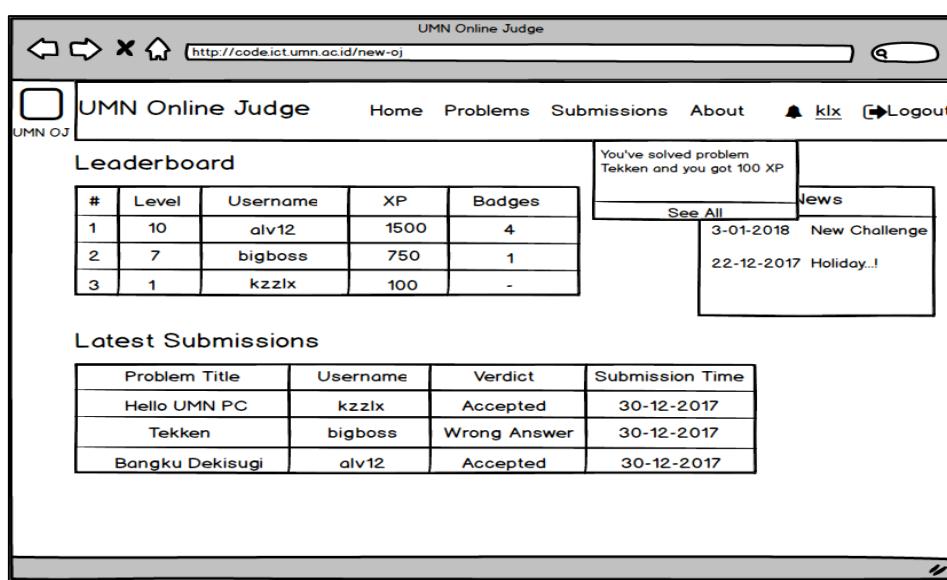
Gambar 3.53 menunjukkan *design user interface invite user*. Tampilan ini merupakan tampilan dari fungsi klik *invite user* pada tampilan *profile*. Tampilan ini merupakan *modal dialog* yang berisi satu *label* yaitu *label email*. Terdapat juga satu *field text box* yang dapat menerima masukan data sesuai dengan *label* yang ada di sampingnya serta tombol Send dan Cancel. Jika tombol Send diklik, maka sistem akan mengirim *email undangan* ke alamat *email* yang telah dimasukkan sebelumnya. Jika tombol Cancel diklik, maka *modal dialog* Invite User akan tertutup.

Gambar 3.54 menunjukkan *design user interface detail news*. Pada tampilan ini, terdapat tiga *label* yaitu *label news*, *news title*, dan waktu *posting news*. Terdapat juga sebuah kotak yang berisi deskripsi dari news yang sedang dibuka.

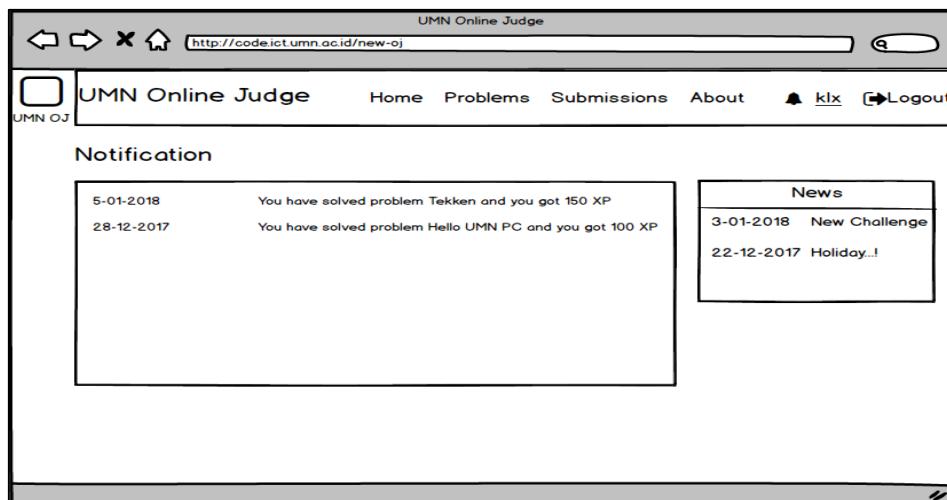


Gambar 3.54 Design User Interface Detail News

Gambar 3.55 menunjukkan *design user interface notification*. Tampilan ini merupakan tampilan *home user*, yang jika *icon notification* diklik akan menampilkan sebuah kotak yang berisi notifikasi yang masuk, serta terdapat *label see all*. Jika *label* tersebut diklik, maka akan masuk ke tampilan all notification.



Gambar 3.55 Design User Interface Notification



Gambar 3.56 Design User Interface All Notification

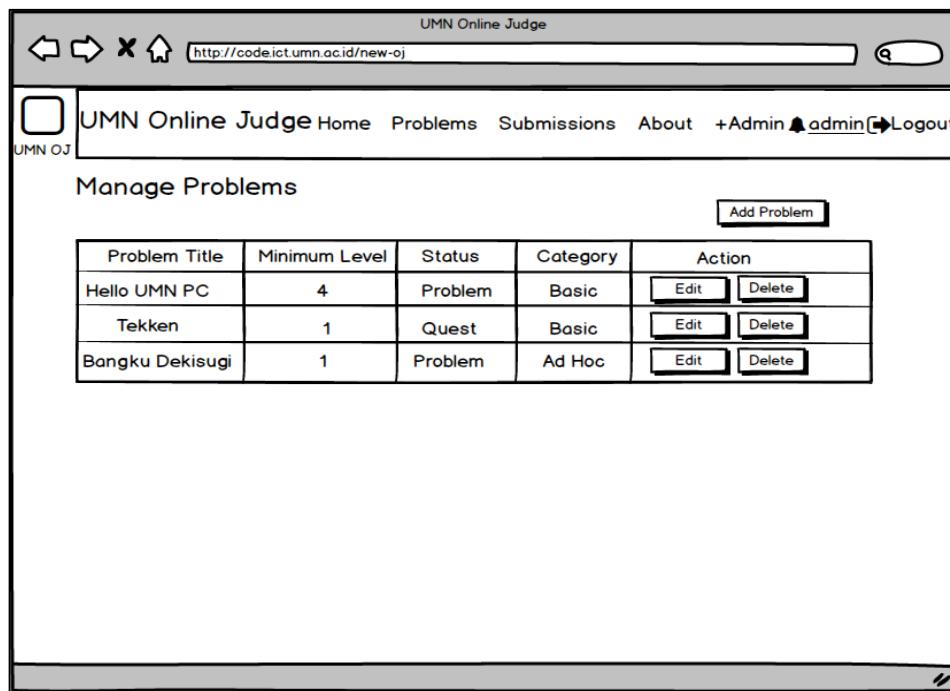
Gambar 3.56 menunjukkan *design user interface all notification*. Pada tampilan ini, terdapat dua *label* yaitu *label notification* dan *news*. Selain itu terdapat *text area* yang berisi semua notifikasi beserta waktu notifikasi masuk. Serta terdapat sebuah kotak berisi *news title* dan waktu *posting*-nya. Jika salah satu dari *news title* diklik, maka akan masuk ke tampilan *detail news*.

#	Level	Username	XP
1	10	alv12	1500
2	7	bigboss	750
3	1	kzzlx	100

Problem Title	Username	Verdict	Submission Time
Hello UMN PC	kzzlx	Accepted	30-12-2017
Tekken	bigboss	Wrong Answer	30-12-2017
Bangku Dekisugi	alv12	Accepted	30-12-2017

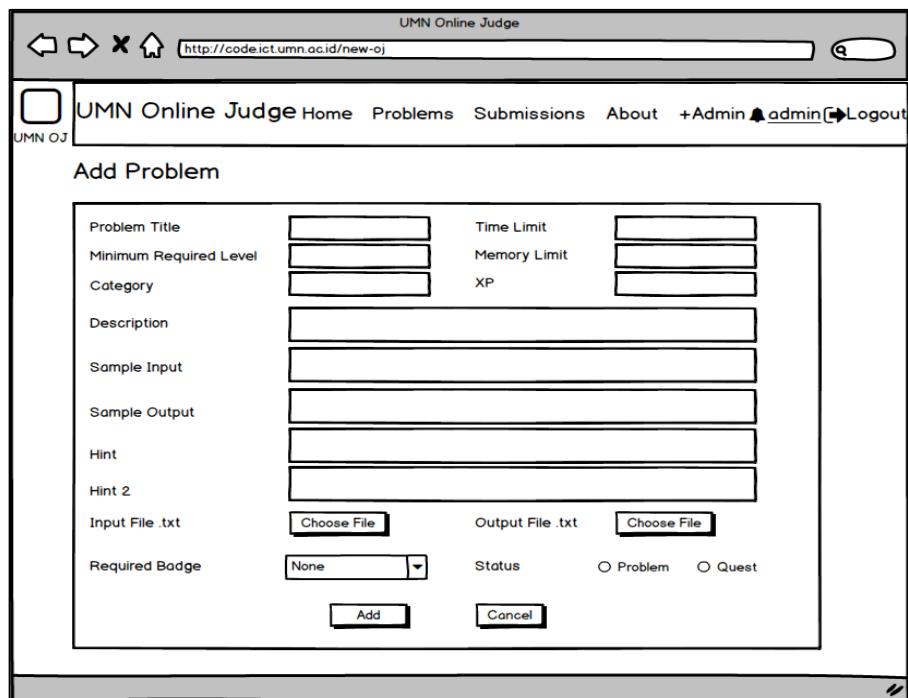
Gambar 3.57 Design User Interface Home Admin

Gambar 3.57 menunjukkan *design user interface home* admin. Pada tampilan ini, terdapat empat *label* yaitu *label leaderboard*, *latest submissions*, *news_username*, dua tabel yaitu tabel *Leaderboard* yang berisi data *level*, *username*, *xp*, dan *badges*. Serta tabel *Latest Submissions* yang berisi data *problem title*, *username*, *verdict*, dan *submission time*. Terdapat pula sebuah kotak berisi *news title* dan waktu *posting*-nya yang jika diklik, akan masuk ke tampilan *detail news*. Selain itu, terdapat juga tab menu seperti menu *Problems*, *Submissions*, dan *Admin*, juga *icon* menu *Logout* dan *Notification*. Jika tab menu *Problem* diklik, maka akan masuk ke tampilan *Problems*. Jika tab menu *Submissions* diklik, maka akan masuk ke tampilan *Submission*. Jika tab menu *About* diklik, maka akan masuk ke tampilan *About*. Jika tab menu *Admin* diklik, maka akan menampilkan sebuah kotak berisi menu tambahan seperti menu *Manage Problem*, *Manage Badge*, *Manage News*, dan *Manage Level*. Jika menu *Manage Problem* diklik, maka akan masuk ke tampilan *manage problem*. Jika menu *Manage Badge* diklik, maka akan masuk ke tampilan *manage badge*. Jika menu *Manage News* diklik, maka akan masuk ke tampilan *manage news*. Jika menu *Manage Level* diklik, maka akan masuk ke tampilan *manage level*. Jika *icon* menu *Notification* diklik, maka akan menampilkan kotak berisi *list notification*. Jika *label username* diklik, maka akan masuk ke tampilan *profile*. Jika *icon* menu *Logout* diklik, maka akan masuk ke tampilan *login*. Jika *label username* diklik, maka akan masuk ke tampilan *profile*.



Gambar 3.58 Design User Interface Manage Problem

Gambar 3.58 menunjukkan *design user interface manage problem*. Pada tampilan ini, terdapat satu *label* yaitu *label manage problems*. Selain itu, terdapat satu tabel yaitu tabel *manage problem* yang berisi data *problem title*, *minimum level*, *category*, dan *action* berupa tombol *Edit* dan *Delete*. Serta terdapat satu tombol yaitu tombol *Add Problem*. Jika tombol *Add Problem* diklik, maka akan masuk ke tampilan *add problem*. Jika tombol *Edit* pada *action manage problem* diklik, maka akan masuk ke tampilan *edit problem*. Jika tombol *Delete* pada *action manage problem* diklik, maka akan masuk ke tampilan *delete problem*.

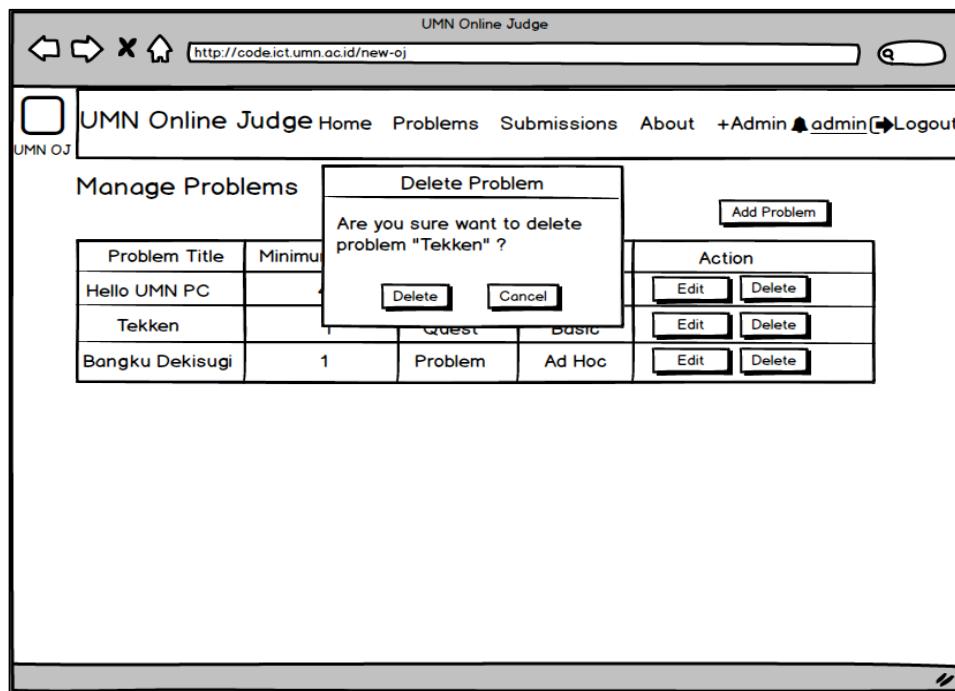


Gambar 3.59 Design User Interface Add Problem

Gambar 3.59 menunjukkan *design user interface add problem*. Pada tampilan ini, terdapat enam belas *label* yaitu *label add problem*, *problem title*, *minimum required level*, *status*, *category*, *time limit*, *memory limit*, *xp*, *description*, *sample input*, *sample output*, *hint*, *hint2*, *input file*, *output file*, dan *required badge*. Terdapat enam *field text box*, lima *field text area* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di samping dan satu *field combo box* yang dapat dipilih nilainya. Terdapat dua *radio button* yang dapat dipilih salah satu nilainya sesuai *label* yang ada di sampingnya. Serta terdapat empat tombol yaitu tombol Choose File untuk *input file*, Choose File untuk *output file*, Add, dan Cancel. Jika tombol Choose file untuk *input file* atau *output file* diklik, maka akan memilih *file input* atau *output*. Jika tombol Add diklik, maka semua data yang telah dimasukkan pada masing-masing *field* serta data *file* yang telah dipilih akan disimpan di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.

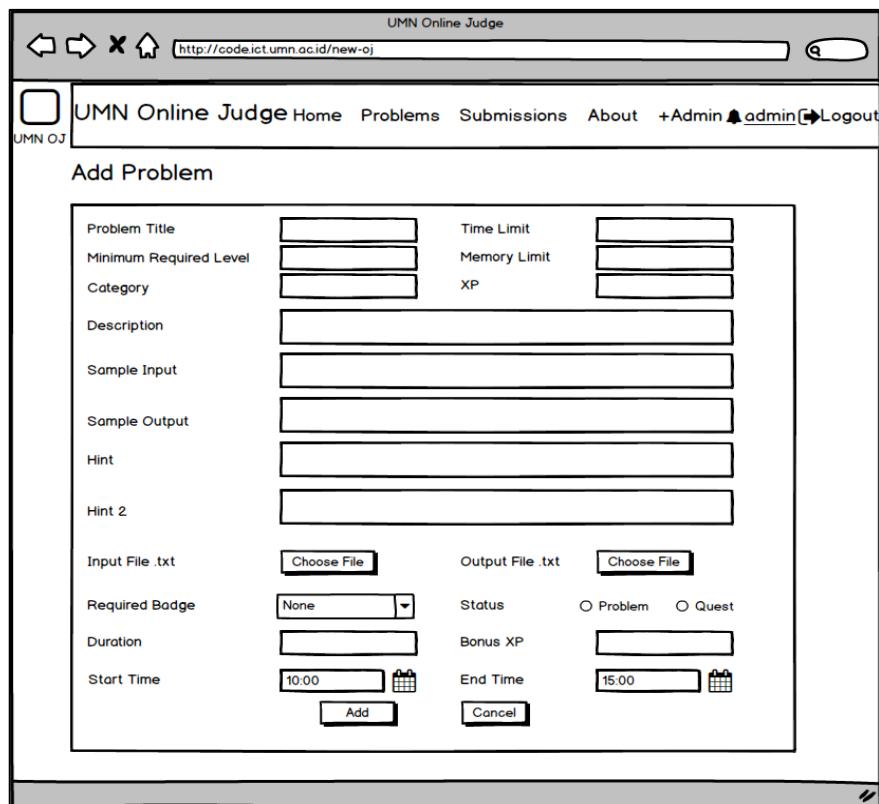
Gambar 3.60 Design User Interface Edit Problem

Gambar 3.60 menunjukkan *design user interface edit problem*. Pada tampilan ini, terdapat enam belas *label* yaitu *label edit problem*, *problem title*, *minimum required level*, *status*, *category*, *time limit*, *memory limit*, *xp*, *description*, *sample input*, *sample output*, *hint*, *hint2*, *input file*, *output file*, dan *required badge*. Terdapat enam *field text box*, lima *field text area* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di sampingnya dan satu *field combo box* yang dapat dipilih nilainya. Terdapat dua *radio button* yang dapat dipilih salah satu nilainya sesuai *label* yang ada di sampingnya. Serta terdapat empat tombol yaitu tombol Choose File untuk *input file*, Choose File untuk *output file*, Edit, dan Cancel. Jika tombol Choose file untuk *input file* atau *output file* diklik, maka akan memilih *file input* atau *output*. Jika tombol Edit diklik, maka semua data yang telah dimasukkan pada masing-masing *field* serta data *file* baru yang telah dipilih akan diperbaharui di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.



Gambar 3.61 Design User Interface Delete Problem

Gambar 3.61 menunjukkan *design user interface delete problem*. Tampilan ini merupakan tampilan dari fungsi *action Delete* pada tabel *manage problem* pada tampilan *manage problem*. Tampilan ini merupakan *modal dialog* yang berisi konfirmasi untuk menghapus suatu *problem*, serta tombol *Delete* dan *Cancel*. Jika tombol *Delete* diklik, maka data *problem* yang dipilih untuk dihapus akan di *soft delete* dari *database* dan *modal dialog* akan tertutup. Jika tombol *Cancel* diklik, maka *modal dialog* akan tertutup.



Gambar 3.62 Design User Interface Add Quest

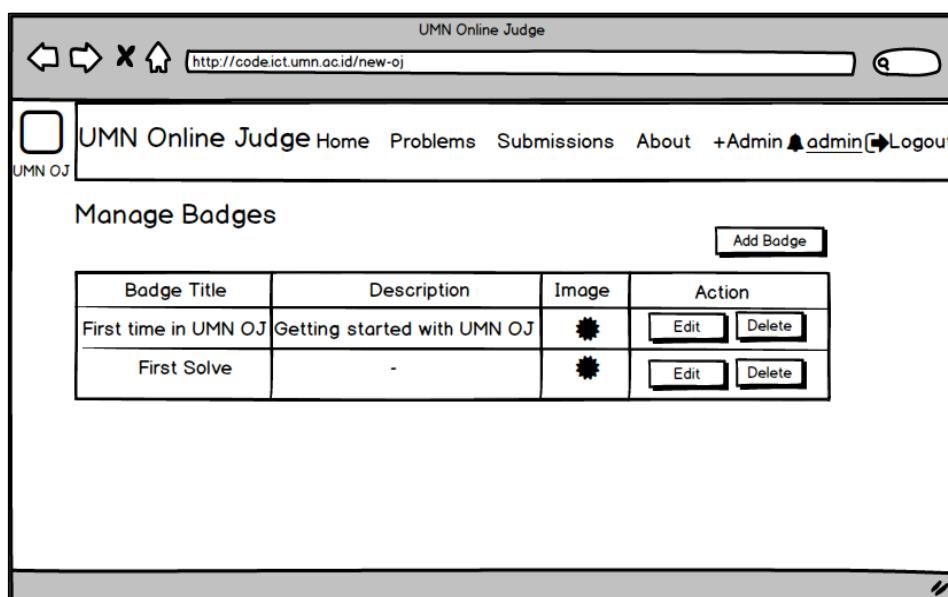
Gambar 3.62 menunjukkan *design user interface add quest*. Pada tampilan ini, terdapat dua puluh *label* yaitu *label add problem*, *problem title*, *minimum required level*, *category*, *duration*, *required badge*, *status*, *time limit*, *memory limit*, *xp*, *bonus xp*, *start time*, *end time*, *description*, *sample input*, *sample output*, *hint*, *hint 2*, *input file*, dan *output file*. Terdapat sepuluh *field text box*, tiga *field text area* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di samping, dua *field datetime* yang dapat dipilih nilainya, dan satu *field combo box* yang dapat dipilih nilainya. Terdapat dua *radio button* yang dapat dipilih salah satu nilainya sesuai *label* yang ada di sampingnya. Serta terdapat empat tombol yaitu tombol Choose File untuk *input file*, Choose File untuk *output file*, Add, dan Cancel. Jika tombol Choose file untuk *input file* atau *output file* diklik, maka akan memilih *file input* atau *output*. Jika tombol Add diklik, maka semua data yang telah

dimasukkan pada masing-masing *field* serta data *file* yang telah dipilih akan disimpan di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.

Gambar 3.63 Design User Interface Edit Quest

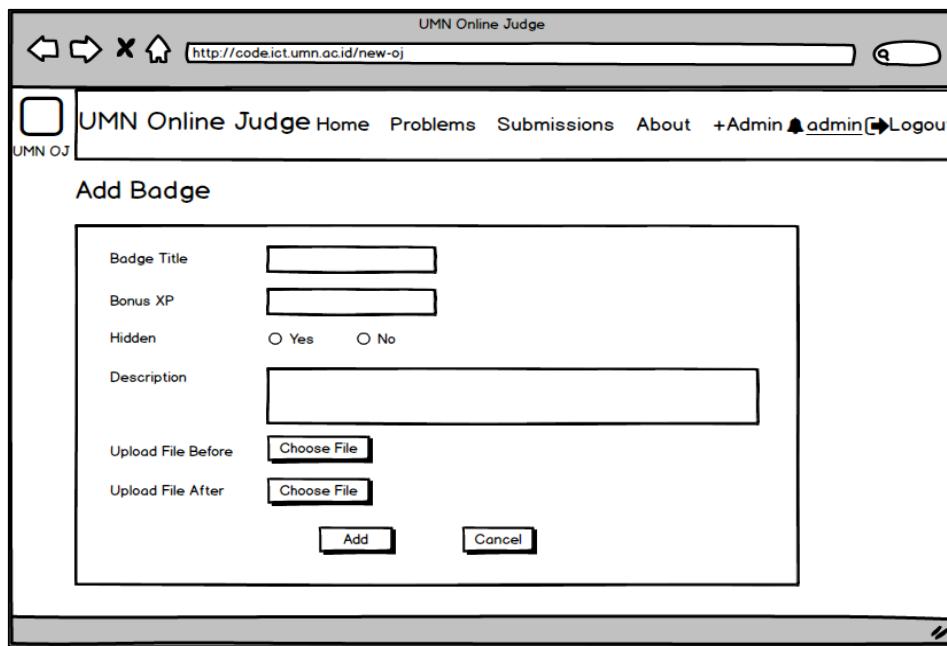
Gambar 3.63 menunjukkan *design user interface edit quest*. Pada tampilan ini, terdapat dua puluh *label* yaitu *label add problem*, *problem title*, *minimum required level*, *category*, *duration*, *required badge*, *status*, *time limit*, *memory limit*, *xp*, *bonus xp*, *start time*, *end time*, *description*, *sample input*, *sample output*, *hint*, *hint 2*, *input file*, dan *output file*. Terdapat sepuluh *field text box*, tiga *field text area* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di samping, dua *field datetime* yang dapat dipilih nilainya, dan satu *field combo box* yang dapat dipilih nilainya. Terdapat dua *radio button* yang dapat dipilih salah satu nilainya sesuai *label* yang ada di sampingnya. Serta terdapat empat tombol yaitu

tombol Choose File untuk *input file*, Choose File untuk *output file*, Edit, dan Cancel. Jika tombol Choose file untuk *input file* atau *output file* diklik, maka akan memilih *file input* atau *output*. Jika tombol Edit diklik, maka semua data yang telah dimasukkan pada masing-masing *field* serta data *file* yang telah dipilih akan disimpan di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.



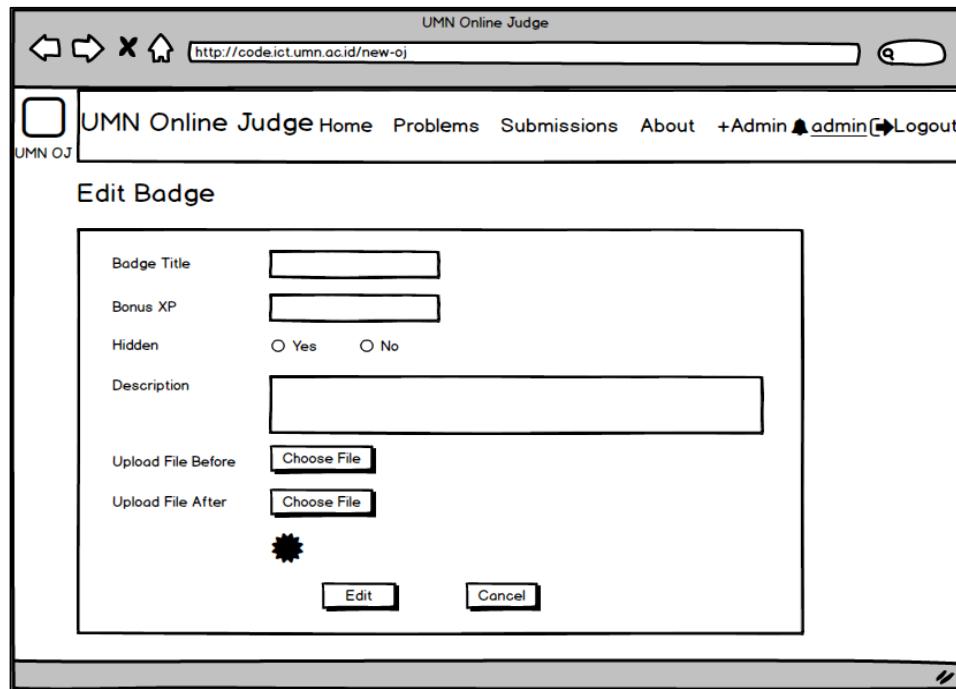
Gambar 3.64 Design User Interface Manage Badge

Gambar 3.64 menunjukkan *design user interface manage badge*. Pada tampilan ini, terdapat satu *label* yaitu *label manage badges*. Selain itu, terdapat satu tabel yaitu tabel *manage badge* yang berisi data *badge title*, *description*, *image*, dan *action* berupa tombol *Edit* dan *Delete*. Serta terdapat satu tombol yaitu tombol *Add Badge*. Jika tombol *Add Badge* diklik, maka akan masuk ke tampilan *add badge*. Jika tombol *Edit* pada *action manage badge* diklik, maka akan masuk ke tampilan *edit badge*. Jika tombol *Delete* pada *action manage badge* diklik, maka akan masuk ke tampilan *delete badge*.



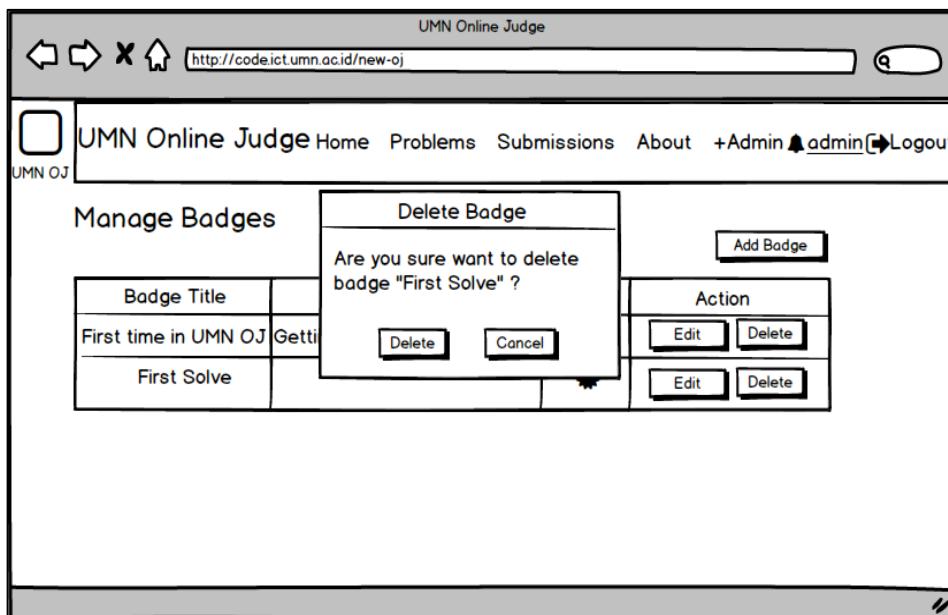
Gambar 3.65 Design User Interface Add Badge

Gambar 3.65 menunjukkan *design user interface add badge*. Pada tampilan ini, terdapat tujuh *label* yaitu *label add badge*, *badge title*, *bonus xp*, *hidden*, *description*, *upload file before*, dan *upload file after*. Terdapat dua *field text box*, satu *field text area*, dua *radio button* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di samping dan dua tombol untuk mengunggah *file* gambar *badge*. Serta terdapat dua tombol yaitu tombol Add dan Cancel. Jika tombol Add diklik, maka semua data yang telah dimasukkan pada masing-masing *field* akan disimpan di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.



Gambar 3.66 Design User Interface Edit Badge

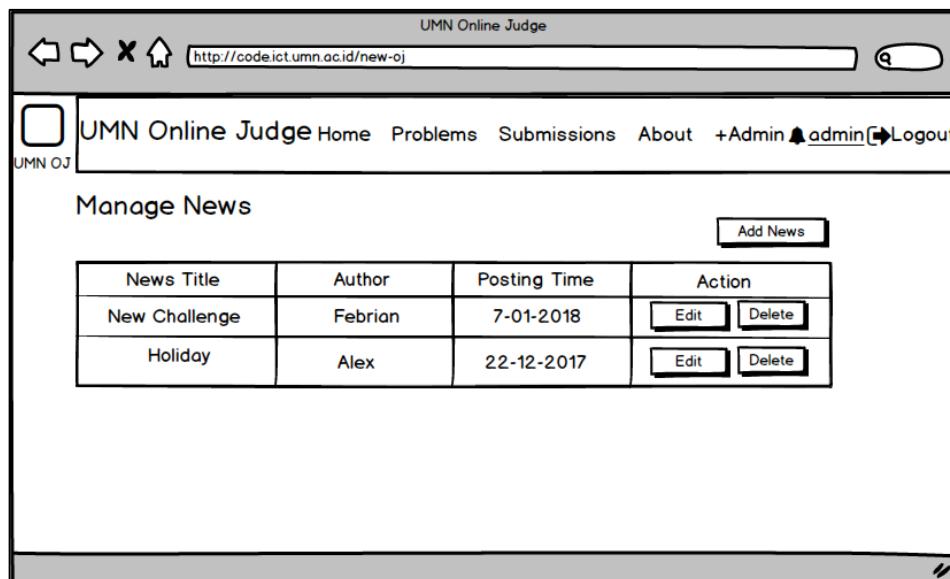
Gambar 3.66 menunjukkan *design user interface edit badge*. Pada tampilan ini, terdapat tujuh *label* yaitu *label add badge*, *badge title*, *bonus xp*, *hidden*, *description*, *upload file before*, dan *upload file after*. Terdapat dua *field text box*, satu *field text area*, dua *radio button* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di samping dan dua tombol untuk mengunggah *file* gambar *badge*. Selain itu, terdapat satu *image* yang berisi gambar *badge* sebelumnya. Serta terdapat dua tombol yaitu tombol *Edit* dan *Cancel*. Jika tombol *Edit* diklik, maka semua data baru yang telah dimasukkan pada masing-masing *field* akan diperbaharui di *database*. Jika tombol *Cancel* diklik, maka semua data yang telah dimasukkan akan di *reset*.



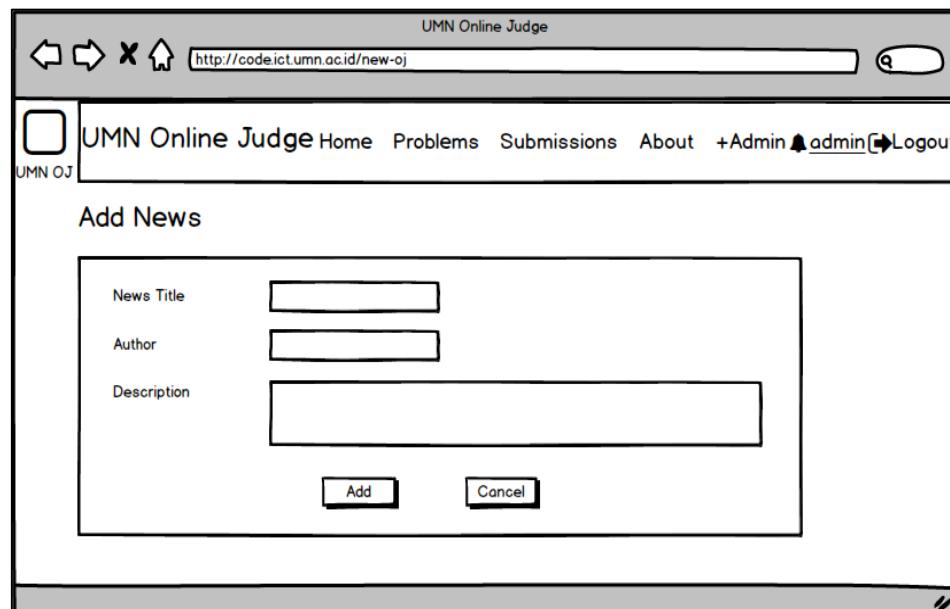
Gambar 3.67 Design User Interface Delete Badge

Gambar 3.67 menunjukkan *design user interface delete badge*. Tampilan ini merupakan tampilan dari fungsi *action Delete* pada tabel *manage badge* pada tampilan *manage badge*. Tampilan ini merupakan *modal dialog* yang berisi konfirmasi untuk menghapus suatu *badge*, serta tombol *Delete* dan *Cancel*. Jika tombol *Delete* diklik, maka data *badge* yang dipilih untuk dihapus akan di *soft delete* dari *database* dan *modal dialog* akan tertutup. Jika tombol *Cancel* diklik, maka *modal dialog* akan tertutup.

Gambar 3.68 menunjukkan *design user interface manage news*. Pada tampilan ini, terdapat satu *label* yaitu *label manage news*. Selain itu, terdapat satu tabel yaitu tabel *manage news* yang berisi data *news title*, *author*, *posting time*, dan *action* berupa tombol *Edit* dan *Delete*. Serta terdapat satu tombol yaitu tombol *Add News*. Jika tombol *Add News* diklik, maka akan masuk ke tampilan *add news*. Jika tombol *Edit* pada *action manage news* diklik, maka akan masuk ke tampilan *edit news*. Jika tombol *Delete* pada *action manage news* diklik, maka akan masuk ke tampilan *delete news*.



Gambar 3.68 Design User Interface Manage News

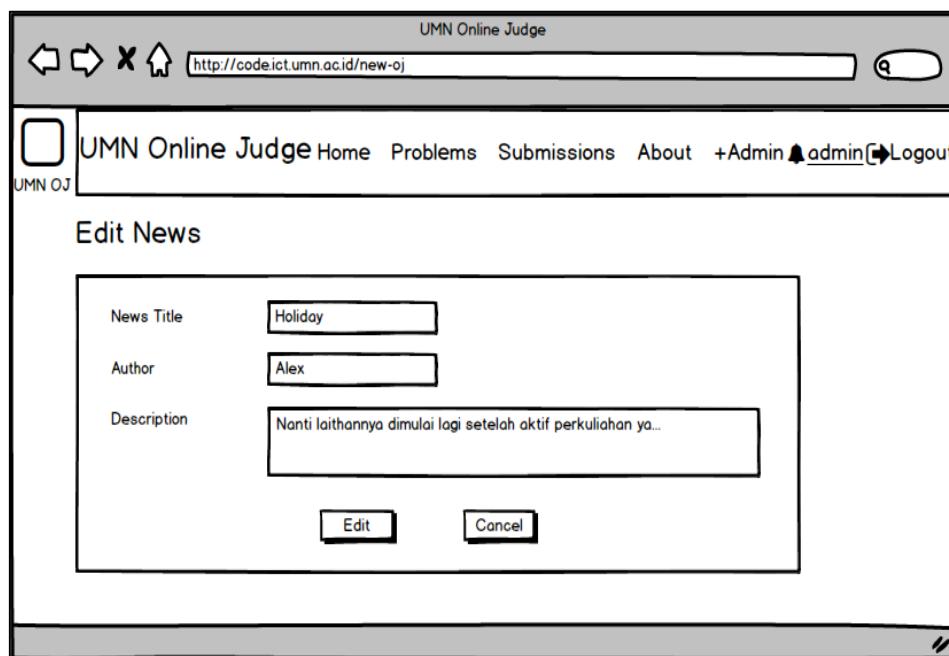


Gambar 3.69 Design User Interface Add News

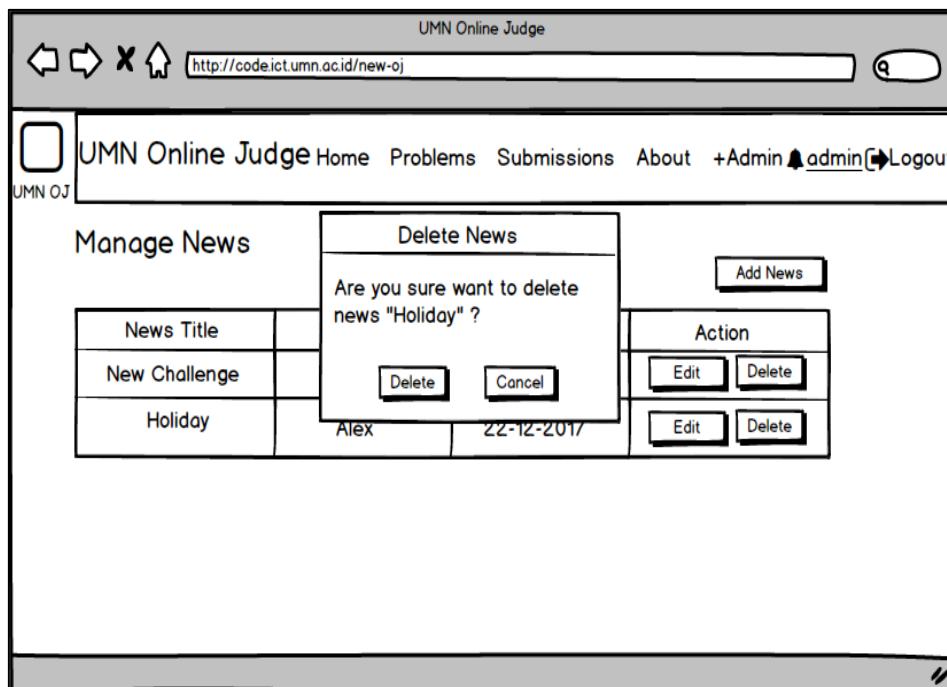
Gambar 3.69 menunjukkan *design user interface add news*. Pada tampilan ini, terdapat empat *label* yaitu *label add news*, *news title*, *author*, *category*, dan *description*. Terdapat dua *field text box*, satu *field text area* yang masing-masing dapat menerima masukan sesuai dengan *label* yang ada di samping. Serta terdapat dua tombol yaitu tombol Add dan Cancel. Jika tombol Add diklik, maka semua data

yang telah dimasukkan pada masing-masing *field* akan disimpan di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.

Gambar 3.70 menunjukkan *design user interface edit news*. Pada tampilan ini, terdapat empat *label* yaitu *label add news*, *news title*, *author*, *category*, dan *description*. Terdapat dua *field text box*, satu *field text area* yang masing-masing dapat menerima masukkan sesuai dengan *label* yang ada di samping. Serta terdapat dua tombol yaitu tombol Edit dan Cancel. Jika tombol Edit diklik, maka semua data baru yang telah dimasukkan pada masing-masing *field* akan diperbaharui di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan akan di *reset*.

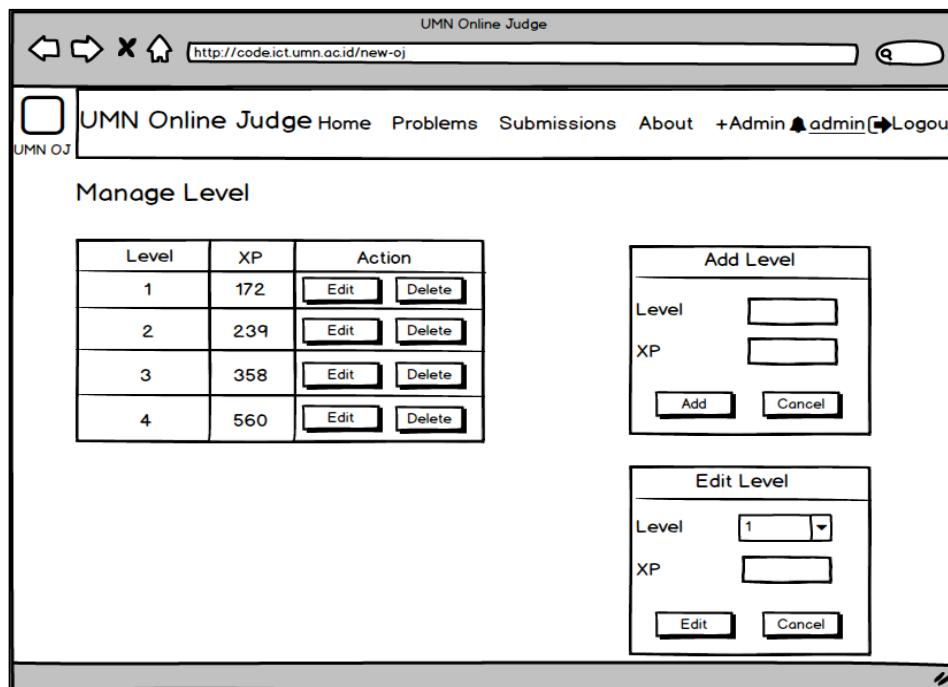


Gambar 3.70 Design User Interface Edit News



Gambar 3.71 Design User Interface Delete News

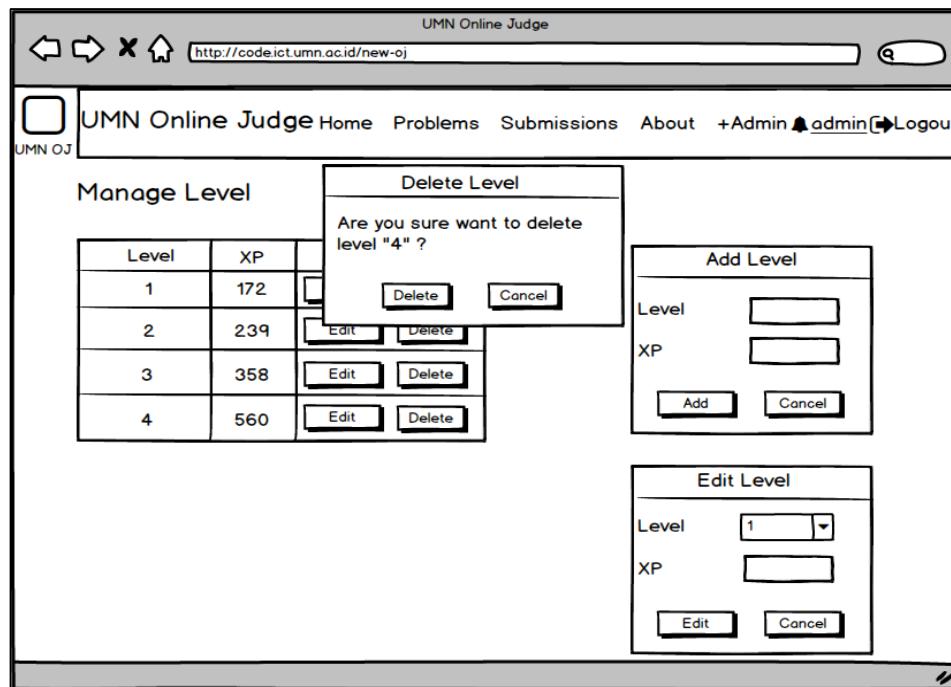
Gambar 3.71 menunjukkan *design user interface delete news*. Tampilan ini merupakan tampilan dari fungsi *action Delete* pada tabel *manage news* pada tampilan *manage news*. Tampilan ini merupakan *modal dialog* yang berisi konfirmasi untuk menghapus suatu *news*, serta tombol *Delete* dan *Cancel*. Jika tombol *Delete* diklik, maka data *news* yang dipilih untuk dihapus akan di *soft delete* dari *database* dan *modal dialog* akan tertutup. Jika tombol *Cancel* diklik, maka *modal dialog* akan tertutup.



Gambar 3.72 Design User Interface Manage Level

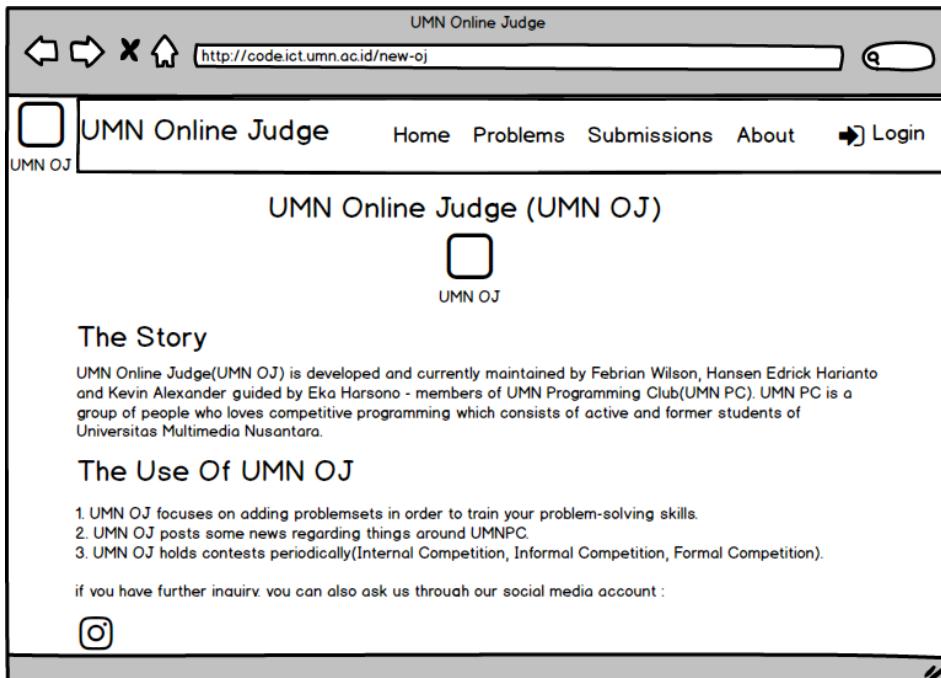
Gambar 3.72 menunjukkan *design user interface manage level*. Pada tampilan ini, terdapat lima *label* yaitu *label manage level*, *level* pada *form Add Level*, *xp* pada *form Add Level*, *level* pada *form Edit Level*, dan *xp* pada *form Edit Level*. Selain itu, terdapat satu tabel yaitu tabel *manage level* yang berisi data *level*, *xp*, dan *action* berupa tombol Delete. Serta terdapat dua *form* yaitu *form Add Level* dan *form Edit Level*. Pada *form Add level* terdapat dua tombol yaitu tombol Add dan Cancel, dan terdapat dua *field text box* yang dapat menerima masukan sesuai *label* di samping. Pada *form Edit level* terdapat dua tombol yaitu tombol Edit dan Cancel, dan terdapat dua *field text box* yang dapat menerima masukan sesuai *label* di samping. Jika tombol Add diklik, maka semua data yang telah dimasukkan pada *field* yang ada akan disimpan di *database*. Jika tombol Cancel diklik, maka semua data yang telah dimasukkan pada *field* yang ada akan di-reset. Jika tombol Edit diklik, maka semua data baru yang telah dimasukkan pada *field* yang ada akan diperbaharui di *database*. Jika tombol Cancel diklik, maka semua data yang telah

dimasukkan pada *field* yang ada akan di-*reset*. Jika tombol Delete pada *action manage level* diklik, maka akan masuk ke tampilan *delete level*.



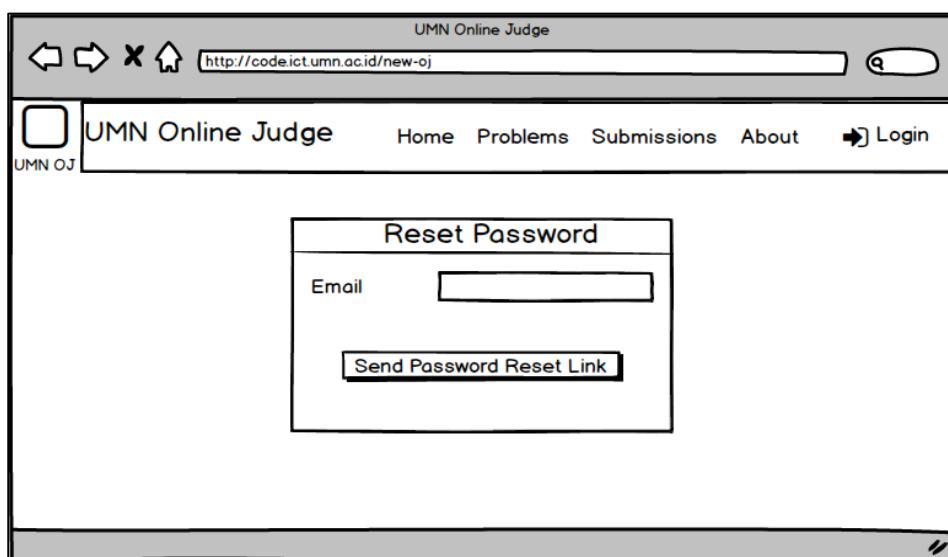
Gambar 3.73 Design User Interface Delete Level

Gambar 3.73 menunjukkan *design user interface delete level*. Tampilan ini merupakan tampilan dari fungsi *action Delete* pada tabel *manage level* pada tampilan *manage level*. Tampilan ini merupakan *modal dialog* yang berisi konfirmasi untuk menghapus suatu *level*, serta tombol Delete dan Cancel. Jika tombol Delete diklik, maka data *level* yang dipilih untuk dihapus akan di *soft delete* dari *database* dan *modal dialog* akan tertutup. Jika tombol Cancel diklik, maka *modal dialog* akan tertutup.



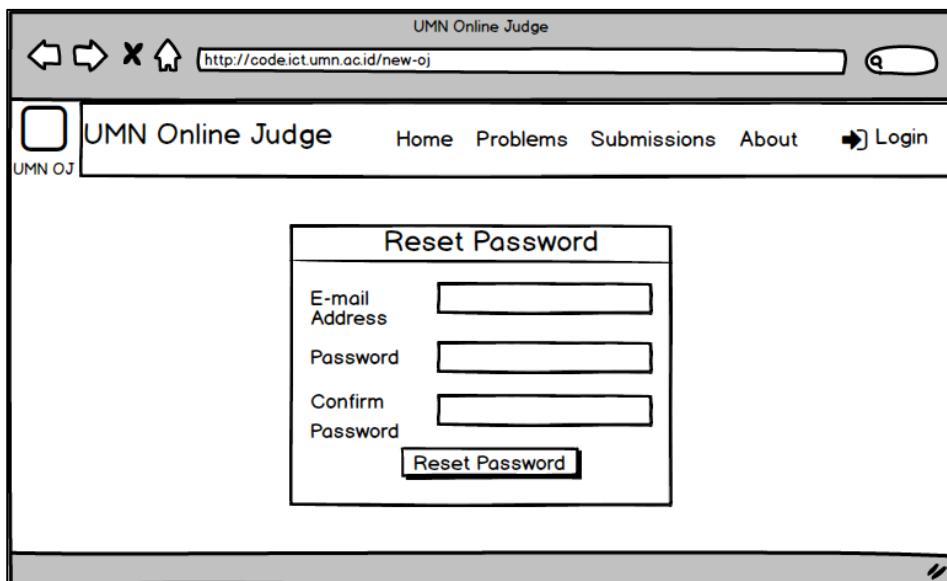
Gambar 3.74 Design User Interface About

Gambar 3.74 menunjukkan *design user interface about*. Pada tampilan ini, terdapat tiga label yang bertuliskan UMN OJ, *the story*, dan *the use of UMN OJ* beserta penjelasannya masing-masing.



Gambar 3.75 Design User Interface Forgot Password

Gambar 3.75 menunjukkan *design user interface forgot password*. Pada laman ini, terdapat satu *label*, satu *field text box*, dan satu *button*. *Text box* dapat menerima masukan yang sesuai dengan *label* di sampingnya.



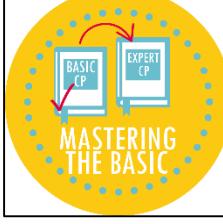
Gambar 3.76 Design User Interface Reset Password

Gambar 3.76 menunjukkan *design user interface reset password*. Pada laman ini, terdapat tiga *label*, tiga *text box*, dan satu *button*. Masing-masing *text box* dapat menerima masukkan sesuai dengan *label* di sampingnya masing-masing.

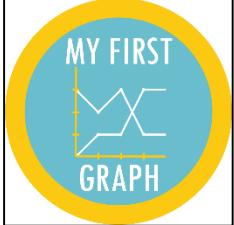
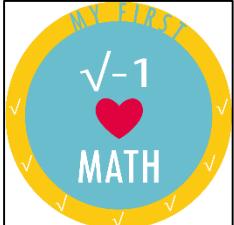
3.3.6 Daftar Aset

Daftar asset *badge* yang digunakan merupakan aset yang dibuat oleh Yuni Ervina Rianto. Tabel 3.15 menunjukan aset *badge* yang digunakan pada UMN Online Judge.

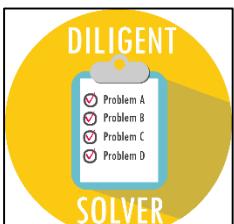
Tabel 3.15 Daftar Aset Badge UMN OJ oleh Yuni Ervina Rianto

No.	Gambar	Keterangan
1.	 Welcome to UMN OJ	First Time in UMN OJ
2.	 First Time Accepted	First time to solve a problem
3.	 Pass Introduction	Successfully pass the input/output material, iterations, and conditions
4.	 Quest Solver	Successfully done three out of five quests
5.	 Mastering the basic	Successfully through the basic materials

Tabel 3.15 Daftar Aset Badge UMN OJ oleh Yuni Ervina Rianto (Lanjutan)

No.	Gambar	Keterangan
6.	 One Shot AC	Get accepted on twenty problems on the first attempt
7.	 One Shot AC Streak	Get accepted on fifteenth problems on the first attempt in a row
8.	 My First Graph	First time solve a graph problem
9.	 My First Math	First time solve a math problem
10.	 My First String	First time solve a string problem

Tabel 3.15 Daftar Aset Badge UMN OJ oleh Yuni Ervina Rianto (Lanjutan)

No.	Gambar	Keterangan
11.	 Solve Ten Problems	Successfully solve ten problems.
12.	 Solve Twenty-five Problems	Successfully solve twenty-five problems.
13.	 Solve Fifty Problems	Successfully solve fifty problems.
14.	 Diligent Solver	Successfully solve at least one problem per day in a week
15.	 Try Hard	Successfully solved the problem after more than five times failed attempt

Tabel 3.15 Daftar Aset Badge UMN OJ oleh Yuni Ervina Rianto (Lanjutan)

No.	Gambar	Keterangan
16.	 The Explorer	Has three hidden badges
17.	 The Fighter	Ten failed attempt on the same problem.
18.	 The Socializer	Give comment on twelve problems
19.	 The Achiever	Has XP more than 17000
20.	 My First Quest	First time solve quest