

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Gambaran Permasalahan

3.1.1 Permasalahan

Kondisi global belakangan ini telah memberikan tantangan terhadap dunia pendidikan. Dengan instruksi pemerintah yang dikeluarkan dan membatasi kegiatan pendidikan secara tatap muka, Pembatasan kegiatan tatap muka ini mengakibatkan banyak kegiatan praktikum yang tidak dapat dilaksanakan secara optimal. Hal ini berakibat pada keterlambatan siswa dalam menyerap materi pembelajaran yang diperlukan. Salah satu alternatif terhadap permasalahan ini adalah dengan penggunaan bantuan dari sebuah virtual lab.

Virtual lab merupakan sebuah aktivitas berbasis komputer, dimana pelajar berinteraksi dengan perangkat laboratorium melalui komputer. Dalam penerapannya, kebanyakan virtual lab mengandalkan kemampuan komputasi untuk melakukan simulasi atas apa yang mungkin dilakukan didalam sebuah laboratorium yang diprogram. Namun, terlepas dengan berbagai kelebihananya, penggunaan virtual lab relatif terbatas karena pengembangannya yang membutuhkan biaya. Pada beberapa kegiatan, biaya pengembangan ini akan lebih besar apabila dibandingkan biaya praktikum itu sendiri.

Namun, dengan perkembangan yang ada pada kondisi dunia pendidikan, pengembangan virtual lab menjadi penting karena dapat membantu proses pembelajaran, dengan harapan bahwa penggunaan virtual lab, walaupun tidak

seefektif penggunaan laboratorium fisik, akan masih lebih baik apabila dibandingkan tanpa laboratorium sama sekali. Pada penelitian tugas akhir ini, akan dibuat sebuah virtual lab mengenai perbaikan perangkat PC (*personal computer*).

3.1.2 Pemecahan Masalah

Berdasarkan permasalahan yang ada, maka solusi yang ditawarkan adalah melalui pembuatan sebuah *virtual lab*, untuk digunakan sebagai sebuah media pembelajaran tambahan dalam proses pembelajaran. *Virtual lab* berpotensi dalam meningkatkan efektivitas proses pembelajaran. Dalam penelitian ini, *virtual lab* yang akan dikembangkan adalah sebuah *virtual lab* untuk proses perbaikan perangkat PC (*personal computer*). Dengan sebuah *virtual lab*, maka proses pembelajaran topik yang berkaitan dengan perbaikan komputer diharapkan dapat menjadi lebih efektif dibandingkan apabila tanpa menggunakan bantuan *virtual lab*, walaupun mungkin tidak seefektif apabila pembelajaran dilakukan dalam situasi laboratorium yang sesungguhnya.

3.2 Analisa

3.2.1 Definisi Kebutuhan Sistem

Dengan menggunakan pendekatan *waterfall*, maka langkah paling awal adalah definisi kebutuhan sistem. Pendefinisian kebutuhan sistem ini didasarkan pada kebutuhan umum atas sistem yang akan digunakan. Pada penelitian ini sistem yang akan dikembangkan adalah suatu aplikasi *Virtual Lab*, yang mana pengembangan ini akan disusun dengan mengikuti definisi kebutuhan sistem dasar sebagai berikut:

Tabel 3.1 Definisi Kebutuhan Sistem

No	Fitur	Keterangan
1	Kebutuhan Fungsional	<ul style="list-style-type: none"> - Aplikasi dapat mensimulasikan kerusakan komputer. - Aplikasi dapat mensimulasikan proses perbaikan
2	Kebutuhan Antarmuka	- Aplikasi dapat menerima masukan dari mouse dan keyboard
3	Kebutuhan Kinerja	- Aplikasi harus dapat berjalan tanpa <i>lag</i> lebih dari 200 ms.

3.2.2 *Project Plan*

Rencana proyek pembuatan *virtual lab* ini dijelaskan dalam dua tahapan, yaitu penjabaran Gambaran Umum Sistem, dan Deskripsi Kebutuhan Sistem. Gambaran Umum Sistem menggambarkan mengenai sistem secara umum, yaitu jenis pendekatannya dan tujuan yang diharapkan dari penggunaan pendekatan tersebut. Deskripsi Kebutuhan Sistem akan menetapkan batasan kebutuhan minimal perangkat yang menjalankan sistem ini, agar sistem dapat berjalan dengan baik. Rencana proyek adalah sebagai berikut:

A. Gambaran Umum Sistem

Virtual lab akan dirancang dalam bentuk pendekatan berbasis problem (*problem-based approach*) dimana pengguna akan dihadapkan terhadap lebih dari satu permasalahan. Pengguna akan diarahkan untuk menyelesaikan permasalahan tanpa diberitahukan mengenai akar

permasalahan. Pengguna akan menggunakan kemampuannya dan perangkat diagnostik yang tersedia, untuk mencoba mencari akar permasalahan, sebelum dapat menerapkan solusi yang dibutuhkan. Pendekatan ini diharapkan mampu mengasah kemampuan sekaligus intuisi dari pengguna dalam melakukan perbaikan.

B. Deskripsi Kebutuhan Sistem

Virtual lab ini akan berbentuk sebuah aplikasi yang direncanakan dapat diakses secara daring ataupun secara luring dan dijalankan pada komputer milik pengguna. Untuk memastikan kelayakannya, maka aplikasi ini direncanakan untuk tidak membutuhkan kemampuan komputasi yang besar dari perangkat yang menjalankannya. Direncanakan, Kebutuhan spesifikasi minimal yang dibutuhkan untuk menjalankan *virtual lab* ini adalah sebagai berikut:

- 64 bit cpu intel/amd
- 2GB RAM
- 100 MB Hard disk space
- Direct-X11 video card
- Microsoft Windows 7

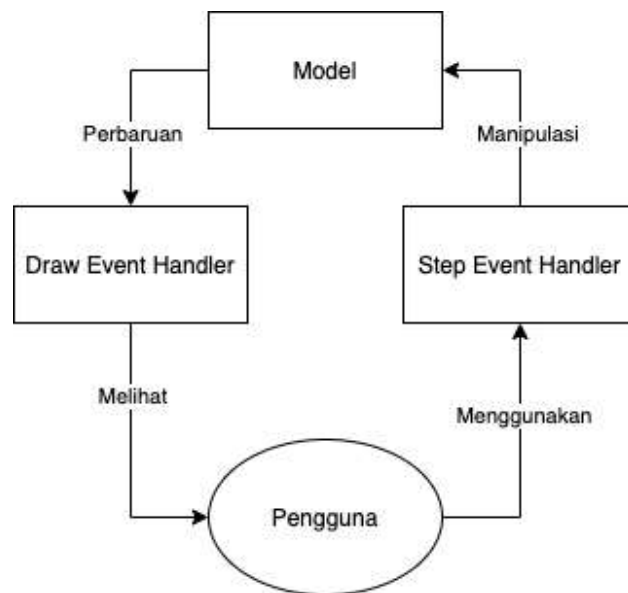
3.2.3 Pembuatan Model

Pembuatan model dilakukan dengan tujuan agar pengembangan *virtual lab* nantinya dapat dirawat, dan memudahkan perbaruan atau penambahan fitur selanjutnya. Pembuatan model ini terdiri dari tiga tahap perancangan, yaitu pembuatan Model Sistem, membuat Deskripsi Pengguna, dan membuat

Pemodelan UML (*Unified Modelling Language*) untuk aplikasi. Tahapan lengkapnya adalah sebagai berikut:

A. Model Sistem

Pola interaksi antar pengguna dan sistem dapat digambarkan dalam sebuah model antar komponen. Layaknya pendekatan berbasis MVC (*model-view-controller*) yang mengatur masing-masing fungsi dalam model, model dalam sistem ini dirancang sebagaimana gambar berikut:



Gambar 3.1 Pembuatan Model Sistem

B. Deskripsi Pengguna

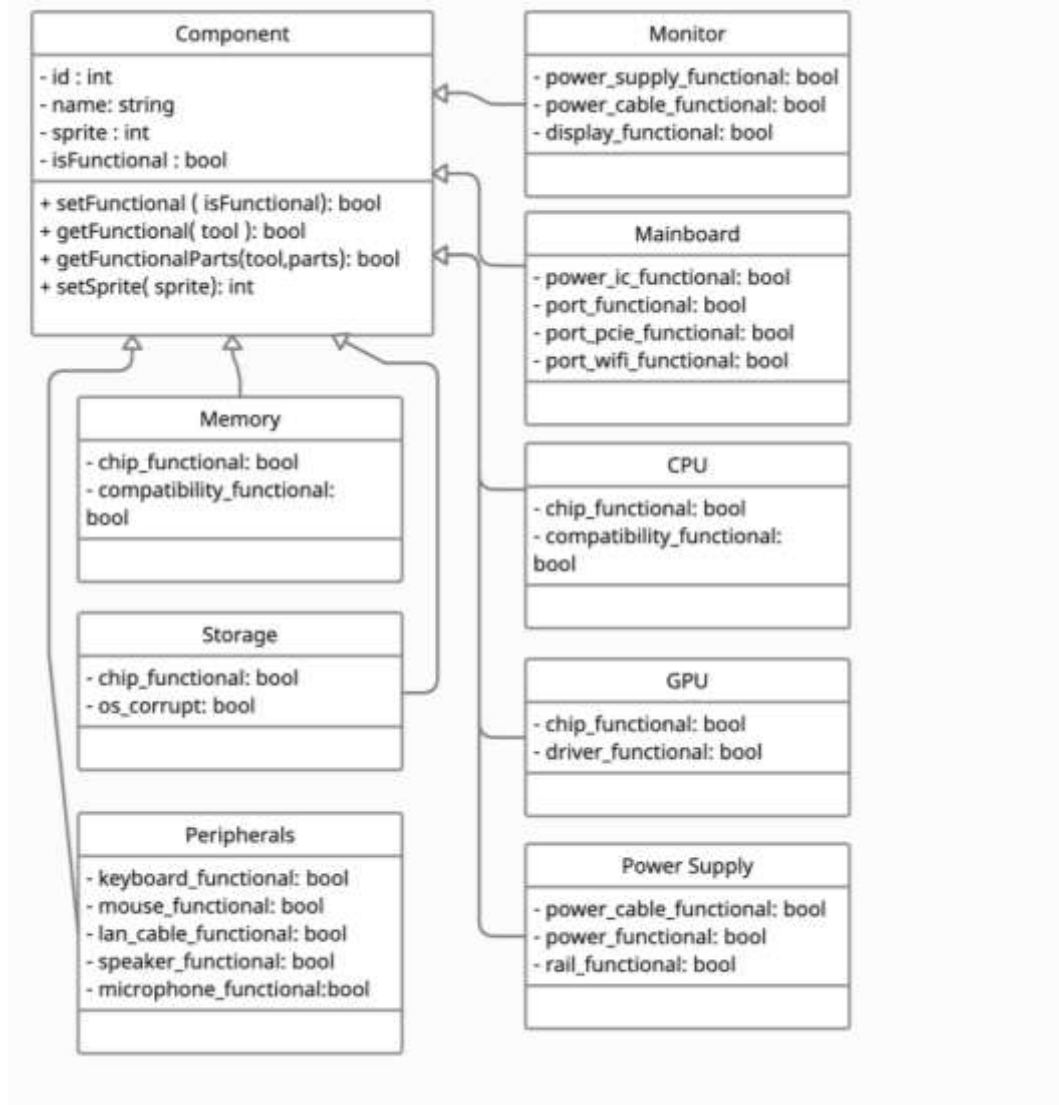
Dalam perancangan sistem dan aplikasi ini, perlu direncanakan suatu profil pengguna yang akan menggunakan aplikasi ini. Pengguna yang diharapkan adalah pelajar atau pengguna yang ingin belajar, atau menguji kemampuan pada *virtual lab* perbaikan komputer ini. Pengguna direncanakan memiliki akses terhadap komputer yang mampu menjalankan program pada *Windows 10*.

C. Pemodelan UML (*Unified Modelling Language*)

Gamemaker studio 2 merupakan IDE yang mendukung penerapan unsur-unsur yang terdapat dalam pemrograman berorientasi obyek (OOP). Salah satu penerapannya adalah kemampuannya untuk membuat sebuah kelas obyek, dan melakukan *inheritance*, dan *overloading*. Untuk itu, dalam pembahasan ini pertama-tama akan dilakukan perancangan kelas. Kelas utama atau *parent* pada penelitian ini adalah sebagai berikut:

1. Kelas komponen
2. Kelas perangkat diagnostik
3. Kelas pengguna

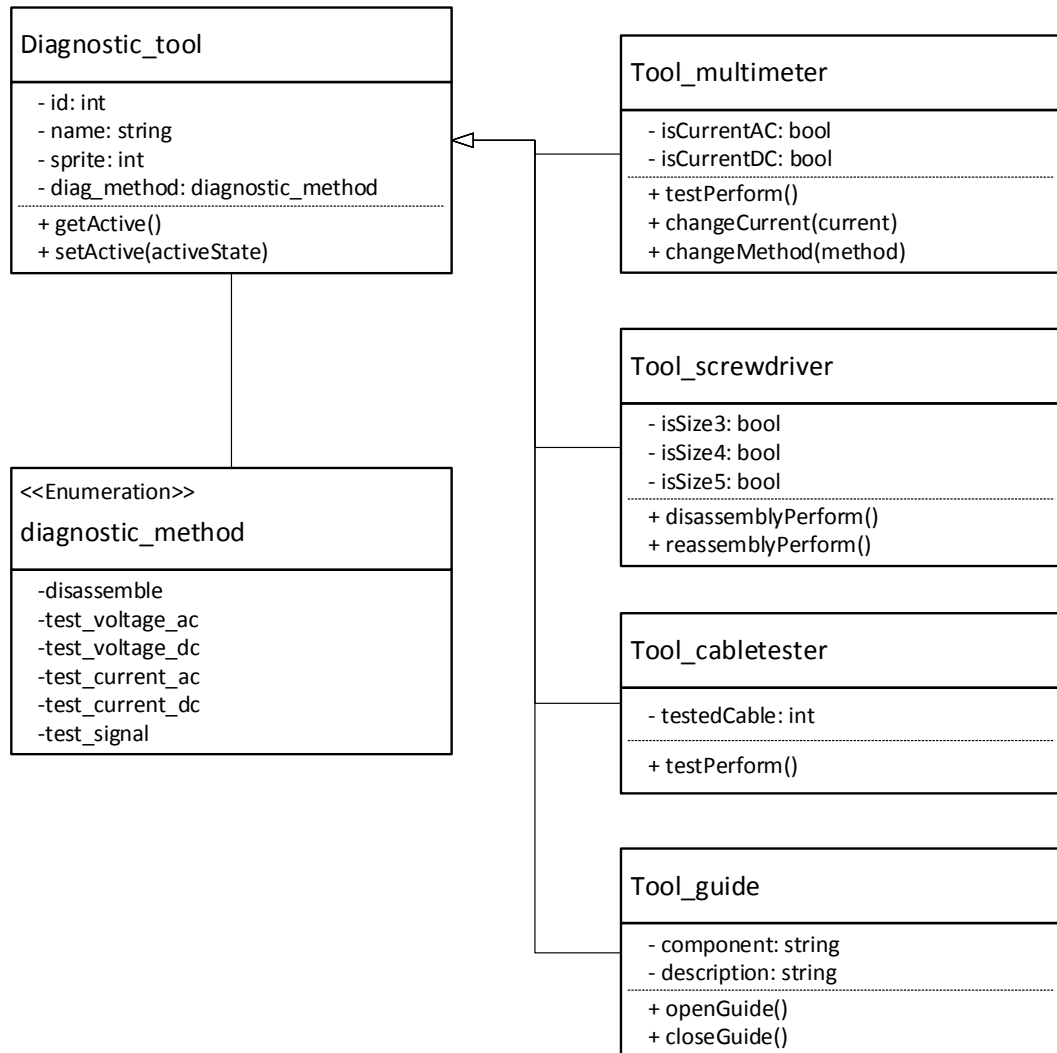
Kelas komponen terdiri dari sub kelas komponen-komponen yang digunakan oleh komputer, untuk dilakukan diagnostik dalam melakukan pencarian kerusakan. Gambar UML class diagram untuk komponen, ditunjukkan pada Gambar 3.2.



Gambar 3.2 UML Class Diagram Kelas Komponen

Kemudian, kelas utama berikutnya adalah kelas perangkat diagnostik. Kelas ini akan merepresentasikan perangkat diagnostik yang akan digunakan dalam aplikasi *virtual lab*. Pada kelas ini terdapat beberapa *child class* antara lain multimeter dan *screwdriver*. dari sebuah *parent class* `Diagnostic_tool`. UML class diagram kelas perangkat diagnostik ini

ditunjukkan pada Gambar 3.3. Kelas terakhir berisi data pengguna, semisal waktu penyelesaian dan nilai penyelesaian.

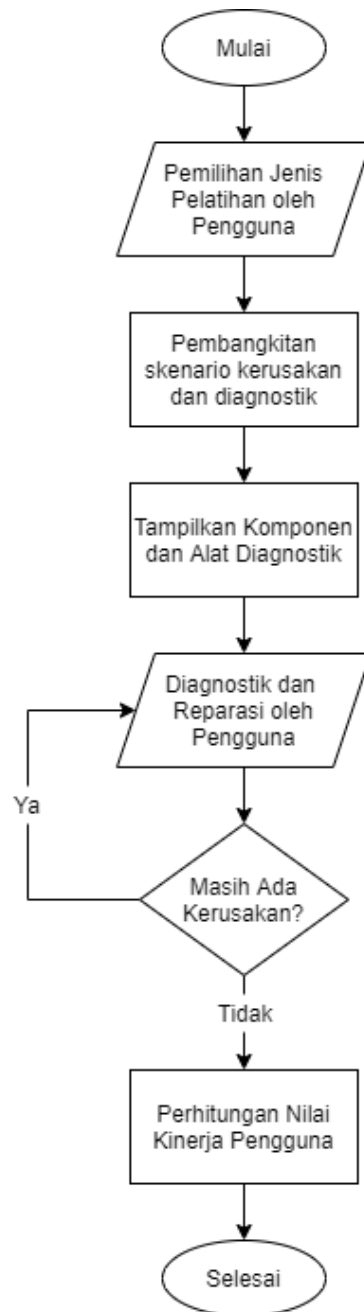


Gambar 3.3 UML Class Diagram Diagnostic Tools

3.3 Perancangan

3.3.1 Desain Sistem dan Aplikasi

Aplikasi virtual lab ini adalah Urutan kerja aplikasi virtual lab ini akan lebih jelas apabila digambarkan dalam sebuah diagram alir (flowchart) aplikasi ini. Diagram alir ini ditunjukkan pada Gambar 3.4.



Gambar 3.4 *Flowchart* Aplikasi *Virtual Lab* Perbaikan Komputer

Pengembangan direncanakan akan dilakukan dengan menggunakan IDE Gamemaker Studio 2. Sebelum itu, perlu dilakukan sebuah perencanaan mengenai virtual lab yang akan dibuat. Pada Tabel 3.1 dijabarkan mengenai alat diagnostic yang tersedia, dan pada Tabel 3.2 dijabarkan mengenai skenario kerusakan yang

mungkin terjadi pada perangkat. Kemudian, dengan menggunakan skenario kerusakan dan alat diagnostik yang akan digunakan, akan disusun diagram alir utama proses pelatihan.

Tabel 3.2 Daftar Perangkat Diagnostik Kerusakan

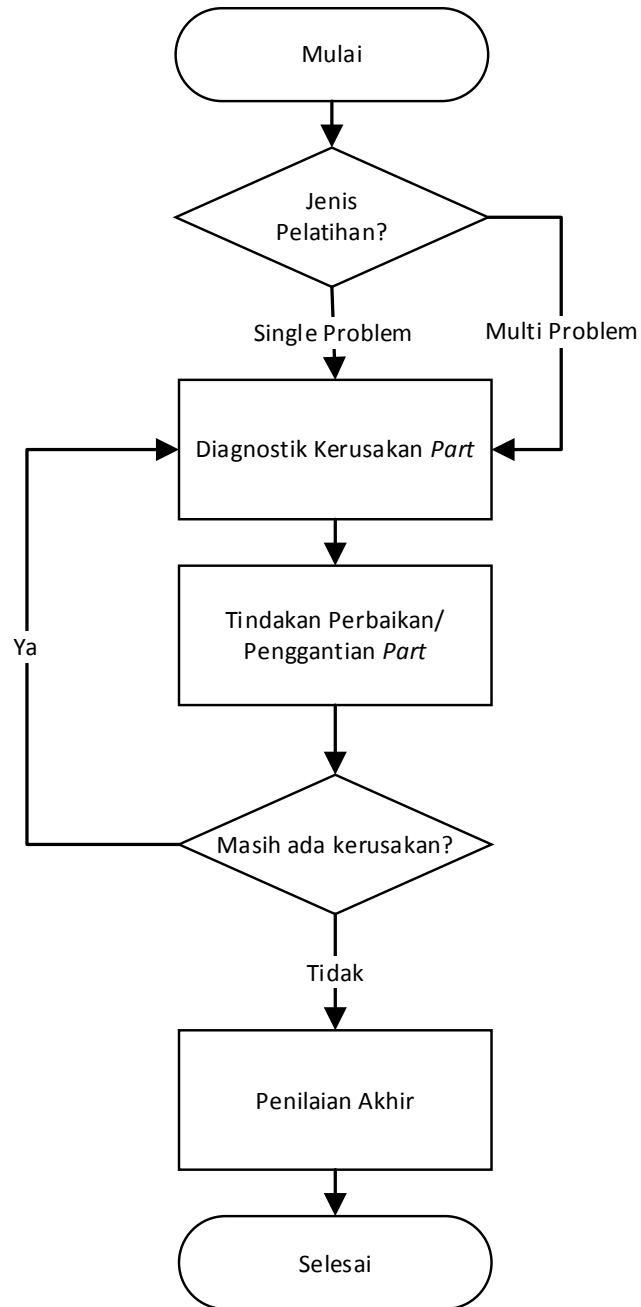
No	Perangkat Diagnostik
1.	Multimeter
2.	<i>Screwdriver Set</i>
3.	Network Repair Toolkit
4.	<i>Spare part komponen</i>
5.	<i>User guide</i>
6.	<i>PC Untuk Diagnostik</i>

Tabel 3.3 Skenario Kerusakan

No	Part	Skenario Kerusakan
1	Monitor	<ul style="list-style-type: none"> - Kerusakan power supply internal monitor - Kerusakan kabel daya - Tampilan monitor tidak sempurna
2	Motherboard/Mainboard	<ul style="list-style-type: none"> - Tidak ada daya listrik yang diterima - Kerusakan port untuk kabel LAN - Tidak ada jaringan WiFi yang diterima
3	CPU	<ul style="list-style-type: none"> - Kerusakan komponen CPU - Ketidak sesuaian komponen CPU dengan <i>mainboard</i>
4	RAM	<ul style="list-style-type: none"> - Kerusakan komponen RAM - Ketidak sesuaian komponen RAM dengan <i>mainboard</i>
5	Harddisk/Storage	<ul style="list-style-type: none"> - Kerusakan komponen harddisk - <i>Corrupt</i> pada system file
6	GPU	<ul style="list-style-type: none"> - Kerusakan GPU - Kerusakan slot mainboard yang digunakan GPU - Kerusakan driver
7	Power Supply	<ul style="list-style-type: none"> - Kerusakan kabel catu daya power supply menuju jala-jala - Power supply tidak mengeluarkan daya sama sekali - Salah satu voltage rail tidak dapat mendukung arus yang dibutuhkan.

8	Peripherals	<ul style="list-style-type: none">- Kerusakan komponen keyboard- Kerusakan komponen mouse- Kerusakan kabel LAN- Kerusakan speaker- Kerusakan microphone
---	-------------	---

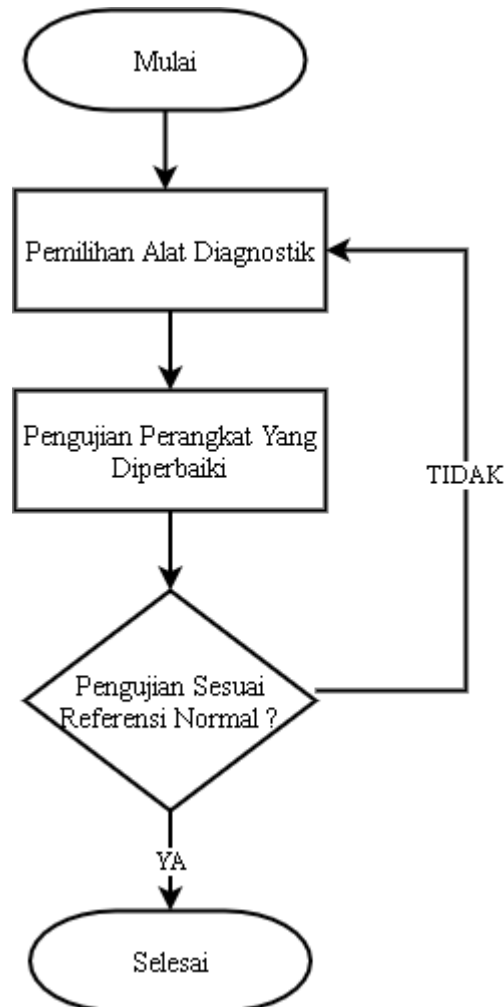
Setelah dilakukan penjabaran skenario kerusakan dan alat diagnostik yang digunakan, maka langkah selanjutnya adalah menyusun diagram alir prosedur kegiatan yang akan dilakukan di dalam virtual lab. Diagram alir tersebut dituangkan kedalam gambar sebagai berikut.



Gambar 3.5 Diagram Alir Penggunaan Pada *Virtual Lab*

Pada proses diagnostik, pengguna akan memilih salah satu dari perangkat diagnostik yang tersedia. Perangkat diagnostik tersebut, kemudian akan digunakan untuk menguji perangkat yang akan diperbaiki. Dari hasil pengujian tersebut, pengguna akan ditunjukkan nilai referensi yang ada. Perbedaan nominal hasil

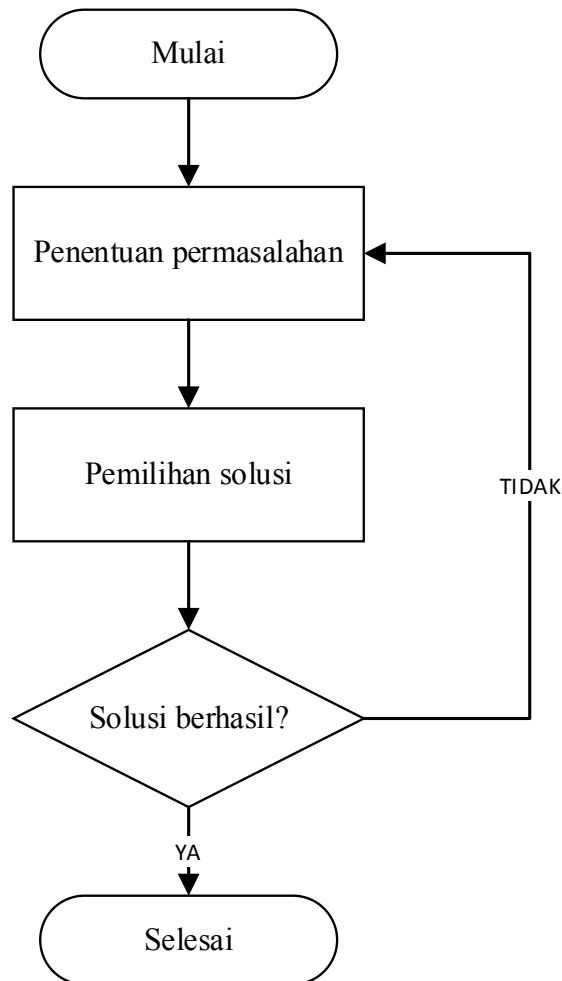
pengujian dan referensi menunjukkan adanya masalah pada perangkat yang diuji tersebut. Diagram alir proses diagnostik ditunjukkan pada Gambar 3.5.



Gambar 3.6 Diagram Alir Proses Diagnostik oleh Pengguna

Setelah potensi permasalahan pada perangkat ditemukan pada proses diagnostik, proses selanjutnya yang dilakukan pengguna adalah implementasi solusi. Pada tahapan ini, pengguna akan diarahkan untuk melakukan tindakan perbaikan yang diperlukan untuk menyelesaikan permasalahan yang telah didiagnostik sebelumnya. Solusi yang tersedia, akan dihubungkan dengan masalah

yang ditetapkan oleh pengguna berdasarkan analisis pada tahapan diagnostik sebelumnya. Diagram alir proses ini ditunjukkan pada Gambar 3.6.



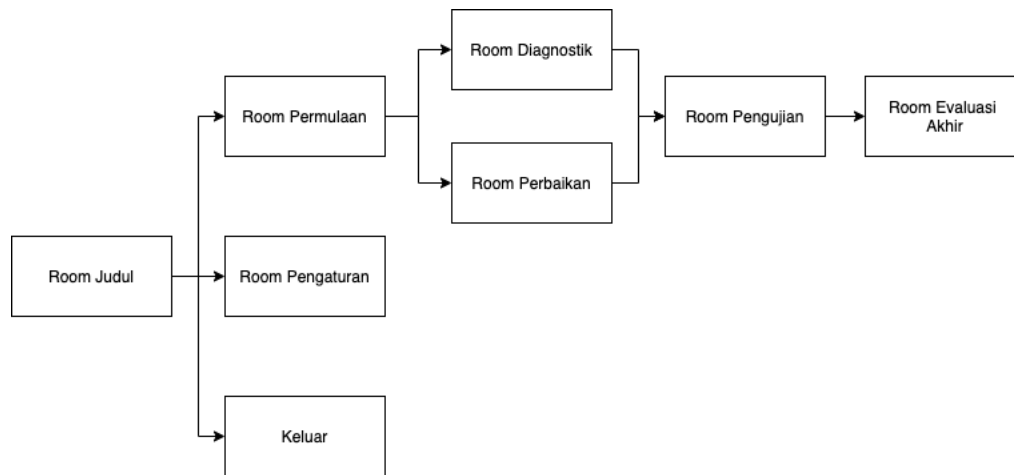
Gambar 3.7 Diagram Alir Proses Implementasi Solusi

Data yang akan digunakan didalam *virtual lab* akan distruktur secara *hard-coded*, sehingga tidak ada data khusus yang disimpan didalam *database*. Hal ini membuat *virtual lab* menjadi lebih cepat dalam pengembangan pada eksperimen yang telah disiapkan secara spesifik ini.

3.3.2 Perancangan Aplikasi

A. Perancangan Struktur Menu Sistem

Struktur menu pada aplikasi ini akan mengarah pada penerapan yang umum digunakan dalam perancangan *game*, dimana struktur merujuk pada perubahan *room*. Struktur menu ini ditunjukkan pada gambar.



Gambar 3.8 Rancangan Struktur Menu Sistem

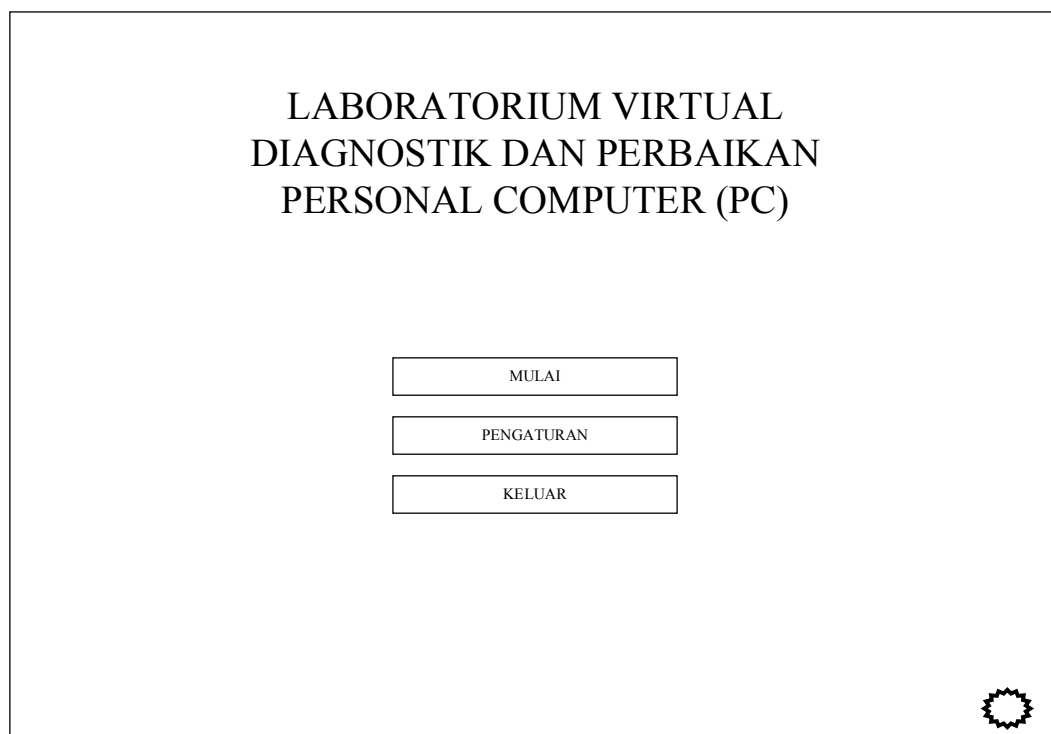
B. Perancangan Antarmuka Pengguna Aplikasi.

Langkah selanjutnya dalam perancangan ini adalah menyiapkan beberapa gambar *mockup* didalam aplikasi. Penggunaan *mockup* ini didasarkan pada *scene* atau *room* (sesuai terminologi Gamemaker Studio 2) yang akan digunakan dalam *virtual lab* ini. Mockup akan terdiri dari :

1. Room judul
2. Room pengaturan
3. Room permulaan
4. Room diagnostik
5. Room perbaikan
6. Room pengujian

7. Room evaluasi akhir

Masing-masing *room* ini akan dijabarkan dengan *mockup* masing-masing untuk memastikan fungsi dan kesesuaian dengan tampilan pada layar *mockup*. *Room* pertama adalah judul. *Room* ini memuat judul dari *virtual lab*, dan beberapa menu pilihan untuk pengguna, yaitu mulai, pengaturan, dan akhiri virtual lab. Pilihan mulai akan mengantarkan pengguna menuju *room* permulaan. Pilihan pengaturan akan mengantar pengguna menuju *room* pengaturan, dan pilihan keluar akan mengakhiri virtual lab. Untuk meningkatkan kualitas visual dari aplikasi *virtual lab*, direncanakan akan dibuat sebuah obyek untuk mengendalikan transisi yang terjadi ketika terjadi pergantian ruangan/*room*.



Gambar 3.9 Mockup *room* judul

Kemudian, dijabarkan mengenai *room* pengaturan. Pada *room* ini akan ditunjukkan mengenai pengaturan terkait *virtual lab* yang akan dilakukan.

Pengaturan ini akan terkait dengan aspek non praktikum, dan aspek praktikum. Pengaturan aspek non praktikum antara lain volume suara, music pengiring, dan aspek grafis terkait. Sementara, aspek praktikum antara lain mode ujian, pemilihan skenario kerusakan, dan lain-lain. Data ini akan disimpan dalam suatu file sehingga pengguna tidak perlu melakukan pengaturan ulang pada pemakaian selanjutnya.

PENGATURAN	
PENGATURAN 1.....	NILAI PENGATURAN 1
PENGATURAN 2.....	NILAI PENGATURAN 2
PENGATURAN 3.....	NILAI PENGATURAN 3
PENGATURAN 4.....	NILAI PENGATURAN 4
PENGATURAN 5.....	NILAI PENGATURAN 5

Gambar 3.10 Mockup *room* pengaturan

Ketika pengguna memencet tombol mulai dari *room* judul, maka pengguna akan dibawa menuju *room* permulaan. *Room* ini digunakan untuk memperkenalkan pengguna terhadap berbagai perangkat diagnostik yang akan digunakan didalam kegiatan tersebut, sekaligus komponen-komponen yang dapat digunakan dalam perbaikan komponen komputer tersebut. *Room* ini bertujuan untuk memberikan gambaran tentang apa yang perlu dilakukan oleh peserta,

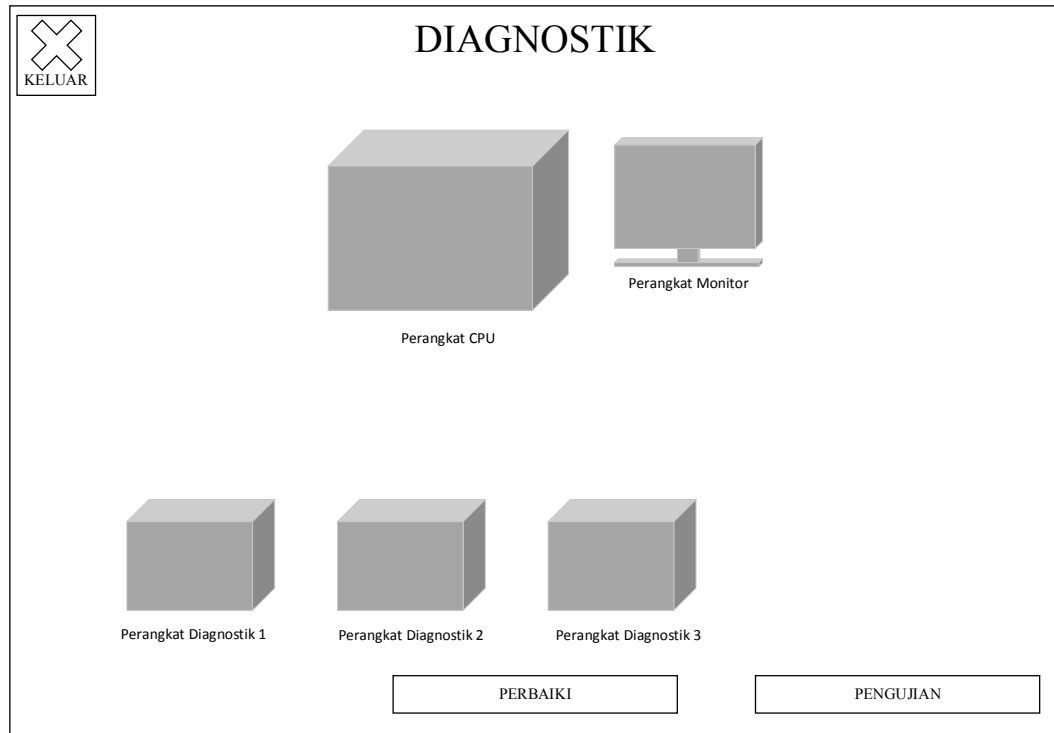
dalam melakukan diagnostik dan perbaikan terhadap sebuah perangkat *personal computer*. Gambar untuk room ini ditunjukkan pada Gambar 3.9.

PERMULAAN

JUDUL PEMBAHASAN
DESKRIPSI PEMBAHASAN

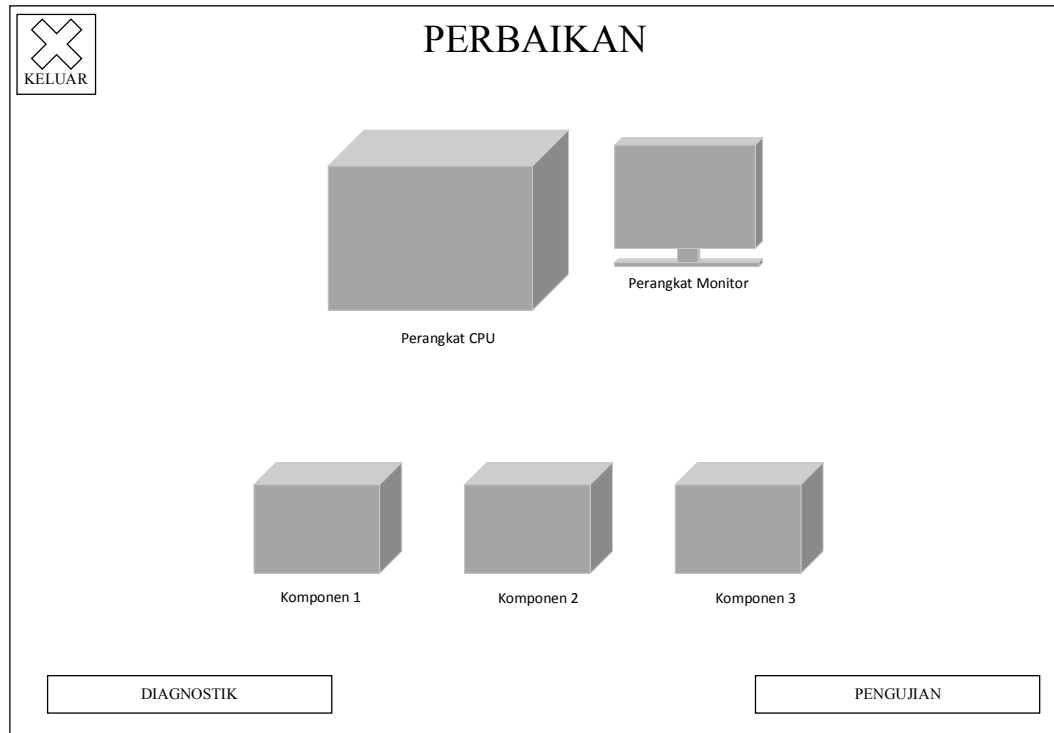
Gambar 3.11 Room permulaan

Setelah melakukan pengenalan dengan berbagai perangkat diagnostik, dan komponen-komponen didalam perangkat *personal computer* (PC), maka pada tahapan selanjutnya, pengguna akan dihadapkan pada permasalahan, yaitu *room* diagnostik. Pada *room* ini, pengguna akan menggunakan peralatan diagnostik yang tersedia, untuk melakukan diagnostik terhadap komponen dari perangkat yang sedang diperbaiki. Berdasarkan data yang diperoleh dari data diagnostik ini, pengguna harus menentukan jenis permasalahan yang dihadapi oleh perangkat yang rusak. *Room* diagnostik ditunjukkan pada Gambar 3.10.



Gambar 3.12 Room diagnostik

Setelah pengguna menemukan potensi permasalahan yang dihadapi, maka pengguna akan memilih "Perbaiki" untuk masuk kedalam *room* perbaikan. Pada *room* perbaikan, akan tersedia metode-metode perbaikan terhadap komponen yang mengalami kerusakan, antara lain dengan melakukan penggantian terhadap komponen yang mengalami kerusakan. Seluruh komponen yang mengalami kerusakan, diasumsikan dapat diganti menggunakan spare part yang tersedia. Cara-cara alternatif semisal *reflow* tidak akan ditunjukkan pada praktikum rancangan ini. Gambar untuk *room* perbaikan ditunjukkan pada Gambar 3.11.



Gambar 3.13 Room perbaikan

Pada skenario *single problem*, maka hanya akan ada satu permasalahan untuk ditemukan. Namun pada skenario *multi problem*, akan ada beberapa kasus yang memiliki permasalahan yang sama. Tergantung dari jenis permasalahannya, maka perangkat PC yang diperbaiki akan mengalami perbaikan kondisi setelah dilakukannya diagnostik. Kemudian pada *room* pengujian akan ditunjukkan bahwa PC dapat kembali bekerja, apabila seluruh permasalahan yang ada telah berhasil diselesaikan oleh pengguna. Pada *room* evaluasi, akan ditunjukkan mengenai permasalahan sebenarnya, beberapa rangkuman, dan efektivitas tindakan yang dilakukan oleh pengguna, sebagai bahan masukan pembelajaran bagi pengguna.