

BAB III

ANALISA DAN PERANCANGAN

III.1. Analisa

Analisa kebutuhan terdapat dua bagian yaitu analisa kebutuhan pengguna dan analisa kebutuhan sistem.

III.1.1. Analisa Kebutuhan Pengguna

Pada analisa kebutuhan pengguna ini, pengguna atau *user* dapat memanfaatkan aplikasi latihan mengetik ini melalui menu yang menjadi antarmuka (*interface*) antara sistem dengan pengguna (*user*). Model penelusuran pemanfaatan aplikasi latihan mengetik ini dengan cara mengetik per-karakter kalimat yang ada pada kotak teks (*textbox*) yang tertera pada rancangan (*design*) aplikasi yang di sediakan sistem. Sistem akan menampilkan waktu dan jumlah kesalahan yang digunakan oleh *user* pada saat *user* selesai mengetik seperti yang tertera pada rancangan (*design*) aplikasi yang disediakan sistem.

III.1.2. Analisa Kebutuhan Sistem

Dalam mengembangkan suatu sistem, diperlukan adanya analisa dan pemodelan terhadap kebutuhan-kebutuhan sistem tersebut, sehingga pada pelaksanaannya sistem tersebut dapat menjalankan proses dengan baik sesuai dengan kebutuhannya.

Analisa kebutuhan pada sistem latihan mengetik ini antara lain :

1. Kebutuhan Aplikasi

Kebutuhan ini meliputi bagaimana sistem dapat menunjang penggunanya dalam mengakses sistem. Kebutuhan aplikasi dari sistem latihan mengetik sebagai berikut :

- a. Sistem yang dikembangkan dapat membantu pengguna dalam hal melatih kemampuan mengetik pengguna tersebut
- b. Sistem yang di rancang akan menampilkan durasi waktu dan jumlah kesalahan yang di lakukan pengguna (*user*) pada saat user telah selesai mengetik sesuai dengan yang tertera pada pada rancangan aplikasi.

2. Kinerja Sistem Yang Diharapkan

Sebuah aplikasi harus dirancang agar dapat menjalankan fungsinya dengan baik, sehingga dapat memberikan solusi atas permasalahan yang ada. Berikut ini merupakan kinerja sistem yang diharapkan dari sistem aplikasi latihan mengetik :

- a. Dukungan *interface* yang dapat memberikan kemudahan bagi pengguna (*user*) dalam menggunakan aplikasi latihan mengetik
- b. Tampilan pada kotak teks (*textbox*) tempat *user* melakukan pengetikan akan berubah warna menjadi merah pada saat *user* melakukan kesalahan pada saat mengetik, hal ini agar mempermudah *user* mengetahui kesalahan pada saat menggunakan aplikasi latihan mengetik.
- c. Sistem akan berhenti pada saat *user* telah selesai melakukan pengetikan seperti yang tertera pada rancangan aplikasi.

III.2. Jenis Perangkat Lunak Yang Di Pakai.

Dalam pengembangan perancangan aplikasi latihan mengetik dengan indikator kesalahan menggunakan fase-fase yang dibutuhkan dalam pengembangan perangkat lunak sehingga hasil akhir akan menghasilkan sistem aplikasi yang strukturnya dapat didefinisikan dengan baik. Adapun instrumen perangkat yang digunakan adalah sebagai berikut:

1. Perangkat Lunak.

Perancangan sistem pendukung menggunakan bahasa pemrograman *VB.Net* 2008, dan sistem operasi *Windows 7*.

2. Perangkat Keras.

Kebutuhan *Hardware* yang digunakan untuk mendukung sistem ini dapat dilihat pada tabel dibawah ini.

Tabel III.1. Kebutuhan hardware

Nama Komponen	Spesifikasi
<i>Procesor</i>	Pentium 4 1.8 Ghz atau lebih tinggi
<i>Memory</i>	1 GB atau lebih tinggi
<i>Harddisk</i>	40 GB atau lebih tinggi
Monitor	10" atau lebih tinggi
<i>Keyboard/Mouse</i>	<i>Standard</i>

III.3. Strategi Pemecahan Masalah

Adapun langkah-langkah yang penulis lakukan untuk merancang aplikasi latihan mengetik adalah sebagai berikut:

1. Mengumpulkan teori-teori aplikasi latihan mengetik.

Dalam tahapan ini, penulis mengumpulkan teori-teori yang berhubungan dengan perancangan aplikasi latihan mengetik. Teori-teori ini penulis kumpulkan

dari beberapa sumber seperti buku-buku di perpustakaan, serta referensi dari beberapa tugas akhir mahasiswa lain yang berhubungan dengan masalah yang dihadapi.

2. Merancang program

Setelah teori-teori penunjang penulis rasakan cukup, langkah selanjutnya penulis melakukan perancangan terhadap program. Program yang dirancang untuk melakukan latihan mengetik.

Langkah pertama dalam perancangan program ini adalah merancang proses kerja sistem. Proses kerja sistem penulis rancang menggunakan sebuah Use Case Diagram yang menjelaskan secara rinci proses-proses yang akan dilakukan program dalam melakukan identifikasi.

Langkah berikutnya adalah merancang bentuk tampilan *form*. Bentuk tampilan *form* yang penulis rancang adalah tampilan *form* dengan menu-menu yang yang dapat digunakan *user* untuk berinteraksi dengan sistem yang dirancang. Dalam langkah ini penulis juga merancang algoritma pemrograman dalam bahasa pemrograman yang digunakan.

3. Mengimplementasikan rancangan program

Bahasa pemrograman yang penulis pilih dalam implementasi rancangan program adalah *Microsoft Visual Studio 2008* atau biasa disebut *VB.Net 2008*.

Pada tahapan ini, penulis mengimplementasikan rancangan tampilan program serta melakukan *coding* sesuai dengan bahasa pemrograman yang digunakan. Tahapan implementasi program yang penulis lakukan adalah membuat

tampilan *form*, membuat *module-module* yang dibutuhkan serta membuat *syntax-syntax* terhadap menu-menu yang ada pada *form*.

Pengimplementasian *form* dan semua fungsi tombol dan menu-menu pada *form* akan di-*check* dan dipastikan apakah sudah berjalan sesuai fungsinya.

4. Melakukan Pengujian Program

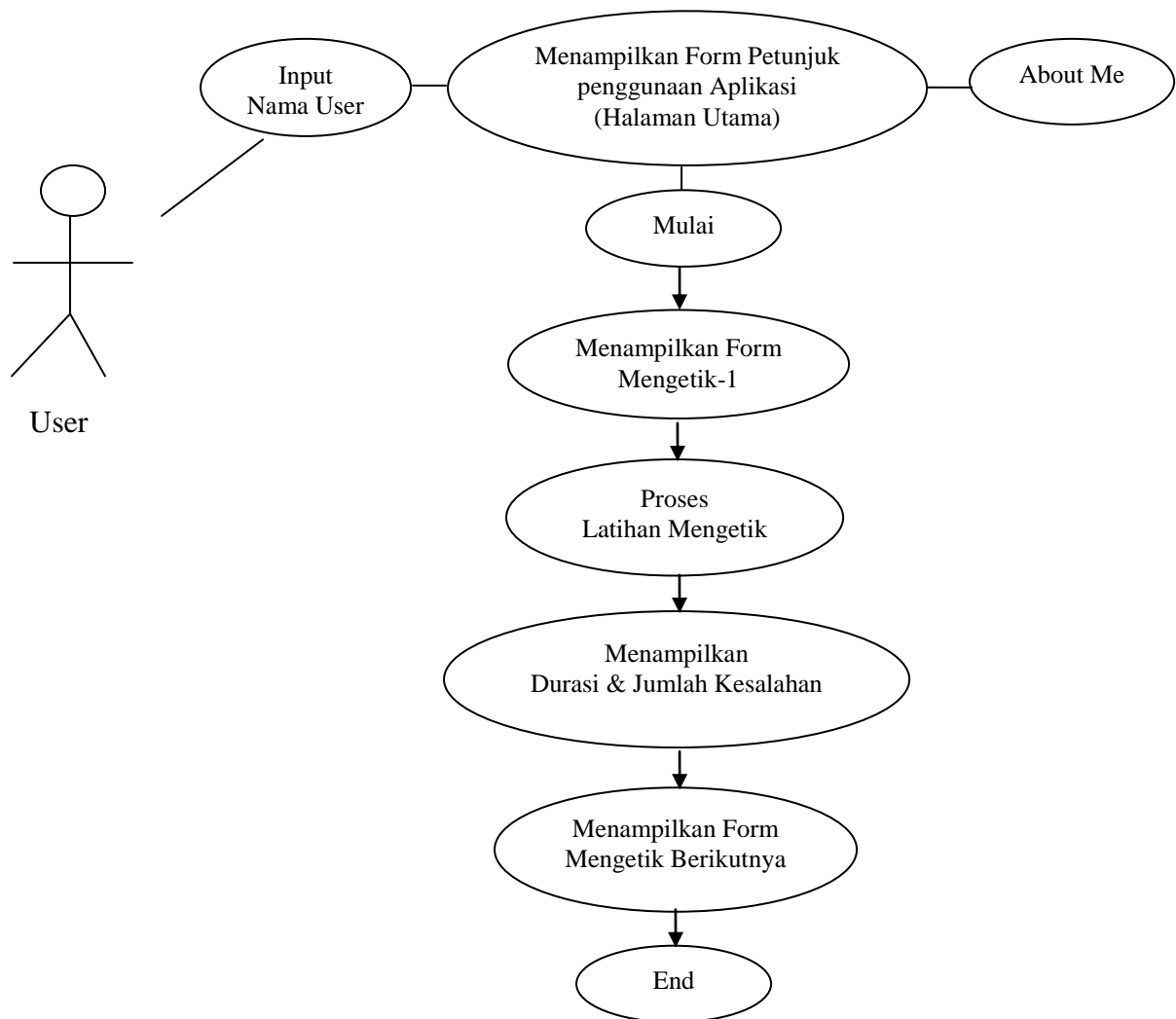
Pada tahapan akhir ini, penulis melakukan serangkaian pengujian terhadap program yang dihasilkan. Pengujian-pengujian ini dilakukan untuk mencari kesalahan-kesalahan (*error*) pada program dan melakukan perbaikan-perbaikan yang dibutuhkan.

III.4. Struktur Data Yang Digunakan

III.4.1. Use Case Diagram

Use Case merupakan permodelan untuk kelakuan (*behavior*) sistem. *Use case* digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem ke pemakai (*user*). *Use case* dapat dilingkupi dengan batasan sistem yang diberi label nama sistem. *Use case* adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.

Use case diagram adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.



Gambar III.1. Use Case Diagram Aplikasi Latihan Mengetik

Ada pun pada Use Case Diagram Aplikasi Latihan Mengetik ini, penulis uraikan penjelasan sebagai berikut:

1. *User* mengisikan identitas diri berupa nama *user*.
2. Setelah *user* meng-*input* nama, kemudian akan tampil *form* petunjuk penggunaan aplikasi (Halaman Utama) yang berisi tentang keterangan penggunaan aplikasi.
3. Kemudian sistem akan menampilkan *form* mengetik-1, *form* ini merupakan area tempat *user* berinteraksi dengan aplikasi yang dirancang.

4. Kemudian proses latihan mengetik, sistem mengidentifikasi per-karakter setiap karakter yang di ketik *user* pada saat menggunakan program.
5. Kemudian sistem akan menampilkan durasi waktu yang digunakan *user* dan jumlah kesalahan yang dilakukan *user* pada saat *user* telah selesai melakukan pengetikan seperti yang tertera pada program.
6. Setelah *user* menyelesaikan pengetikan pada form Mengetik-1, maka sistem akan menampilkan *form* mengetik berikutnya (Mengetik-2, Mengetik-3, Mengetik-4 dan Mengetik-5).
7. Setelah *user* menyelesaikan pengetikan pada *form* Mengetik-5, maka sistem akan menampilkan *form* selesai, yang berarti aplikasi latihan mengetik sudah selesai digunakan.
8. User dapat memilih *About Me* untuk melihat biografi penulis.

III.5. Perancangan

Perancangan adalah pendekatan yang digunakan dalam bidang rekayasa dan bidang lainnya yang digunakan untuk menspesifikasikan bagaimana membuat atau melakukan sesuatu.

Perancangan perangkat lunak menggunakan bahasa pemrograman *Microsoft Visual Basic.Net 2008* dengan menggunakan beberapa komponen *standard visual basic*, seperti: *Label*, *text box*, *PictureBox*, *Timer*, dan *MenuStrip*.

Perangkat lunak memiliki 7 (tujuh) buah *form*, antara lain:

1. *Form* Petunjuk
2. *Form* Mengetik-1

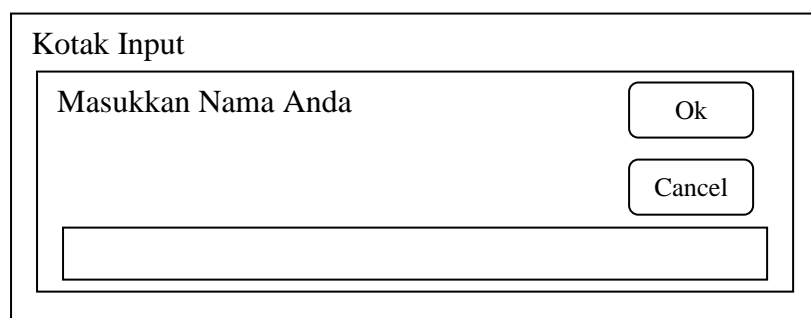
3. *Form Mengetik-2*
4. *Form Mengetik-3*
5. *Form Mengetik-4*
6. *Form Mengetik-5*
7. *Form About Me.*
8. *Form Selesai.*

III.5.1. Rancangan Layar

Pada bagian perancangan ini, penulis melengkapi dengan rancangan *user interface*, yang memudahkan *user* untuk berinteraksi dengan program.

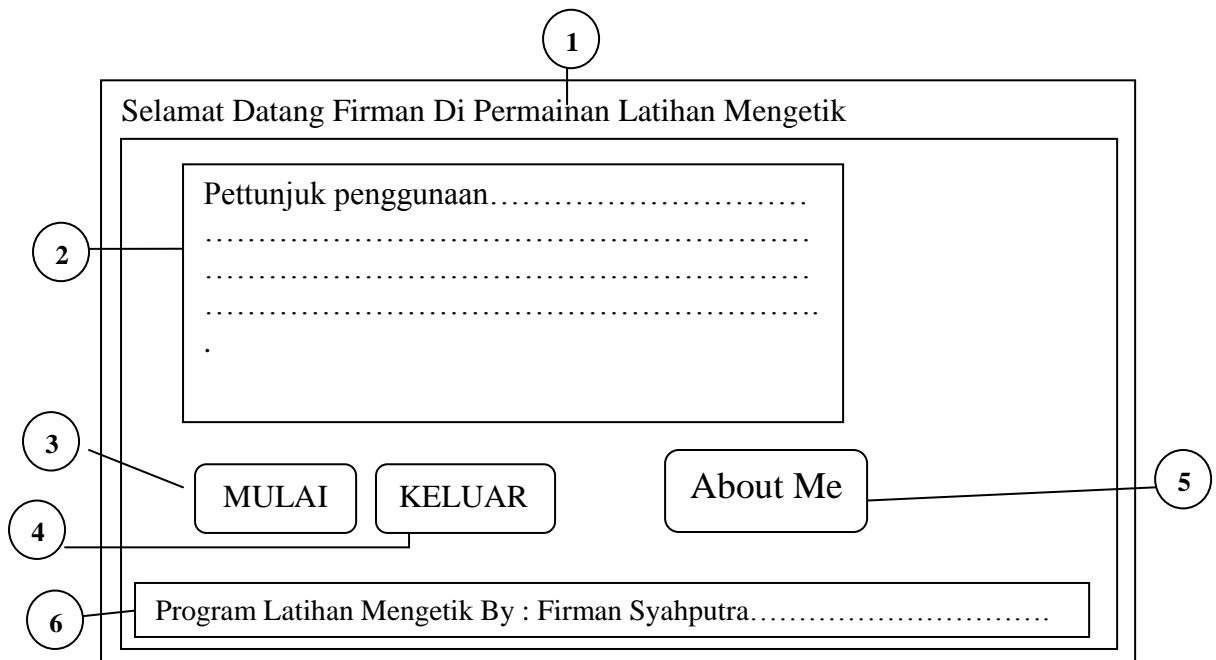
1. *Form Petunjuk (Form Halaman Utama)*

Pada saat program mulai di jalankan, maka sistem akan menampilkan kotak *input* untuk peng-inputan nama *user*, seperti gambar di bawah ini:



The image shows a dialog box titled "Kotak Input". Inside the dialog box, there is a label "Masukkan Nama Anda" followed by a text input field. To the right of the input field are two buttons: "Ok" and "Cancel".

Kemudian setelah *user* meng-input nama, maka sistem akan menampilkan *form* petunjuk seperti berikut :

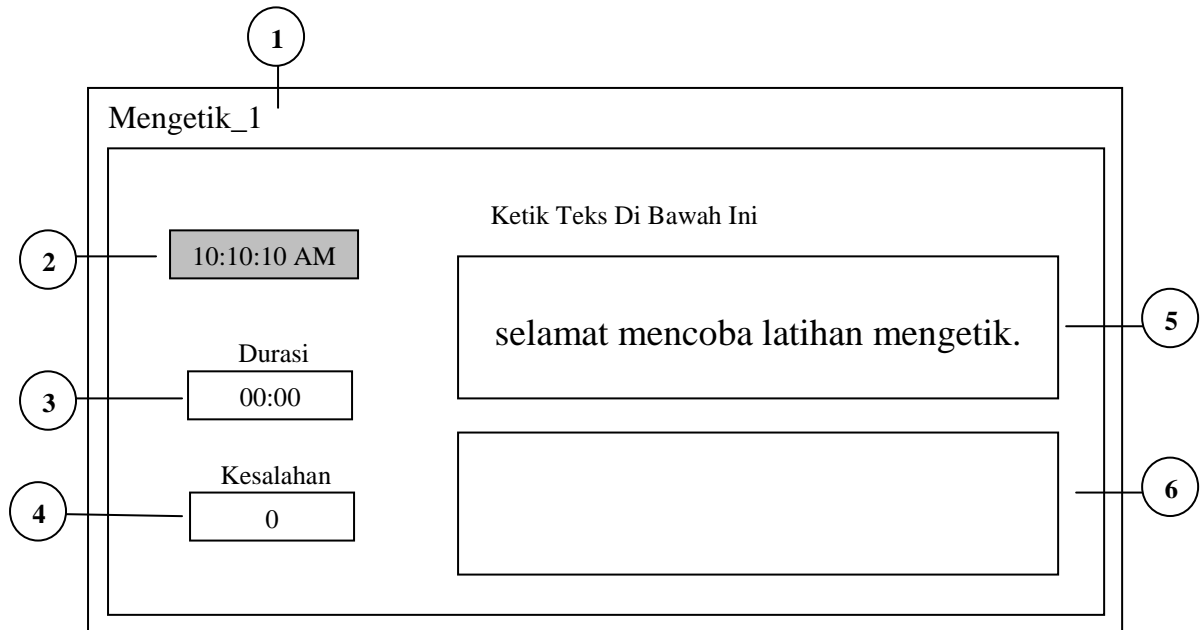


Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title bar</i>
2.	Label	Keterangan petunjuk penggunaan aplikasi
3.	<i>Button</i>	Untuk memulai penggunaan aplikasi, yang akan menampilkan <i>form</i> menetik-1
4.	<i>Button</i>	Untuk keluar dari aplikasi
5.	<i>Button</i>	Untuk menampilkan <i>form about me</i>
6.	<i>Label</i>	Untuk menampilkan <i>running text</i> atau teks berjalan

2. Form Mengetik-1

Adapun *form* mengetik-1 yang dirancang oleh penulis adalah sebagai berikut:

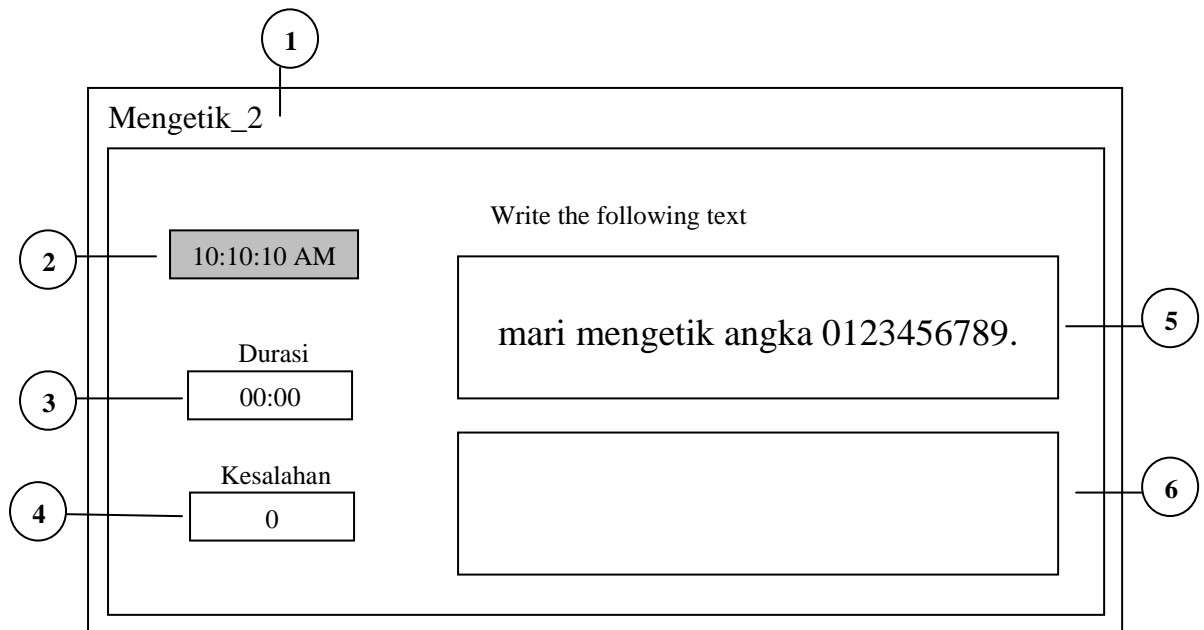


Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title bar</i>
2.	<i>Label</i>	Untuk menampilkan waktu (jam) sesuai dengan format pada PC atau laptop yang digunakan pada saat uji coba program
3.	<i>Label</i>	Untuk menampilkan durasi lamanya waktu yang digunakan <i>user</i> pada saat menggunakan aplikasi
4.	<i>Label</i>	Untuk menampilkan jumlah kesalahan yang dilakukan <i>user</i> pada saat menggunakan aplikasi
5.	<i>Label</i>	Tempat teks yang akan di ikuti
6.	<i>Label</i>	Tempat teks yang akan di ketik

3. Form Mengetik-2

Adapun *form* mengetik-2 yang dirancang oleh penulis adalah sebagai berikut:

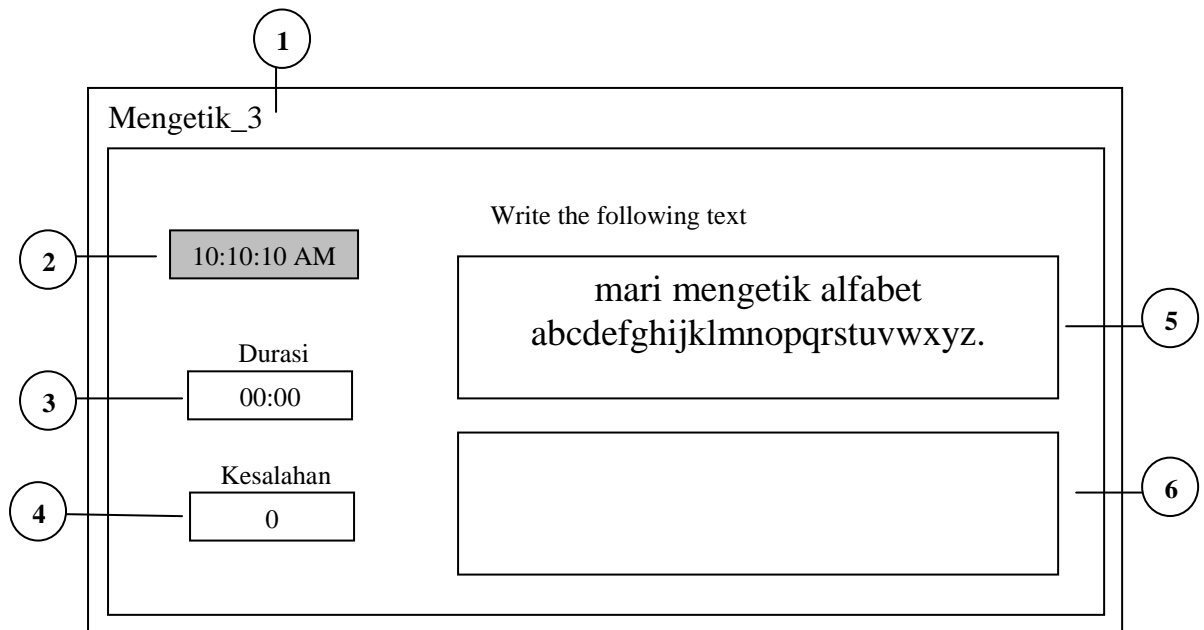


Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title bar</i>
2.	<i>Label</i>	Untuk menampilkan waktu (jam) sesuai dengan format pada PC atau laptop yang digunakan pada saat uji coba program
3.	<i>Label</i>	Untuk menampilkan durasi lamanya waktu yang digunakan <i>user</i> pada saat menggunakan aplikasi
4.	<i>Label</i>	Untuk menampilkan jumlah kesalahan yang dilakukan <i>user</i> pada saat menggunakan aplikasi
5.	<i>Label</i>	Tempat teks yang akan di ikuti
6.	<i>Label</i>	Tempat teks yang akan di ketik

4. Form Mengetik-3

Adapun *form* mengetik-3 yang dirancang oleh penulis adalah sebagai berikut :

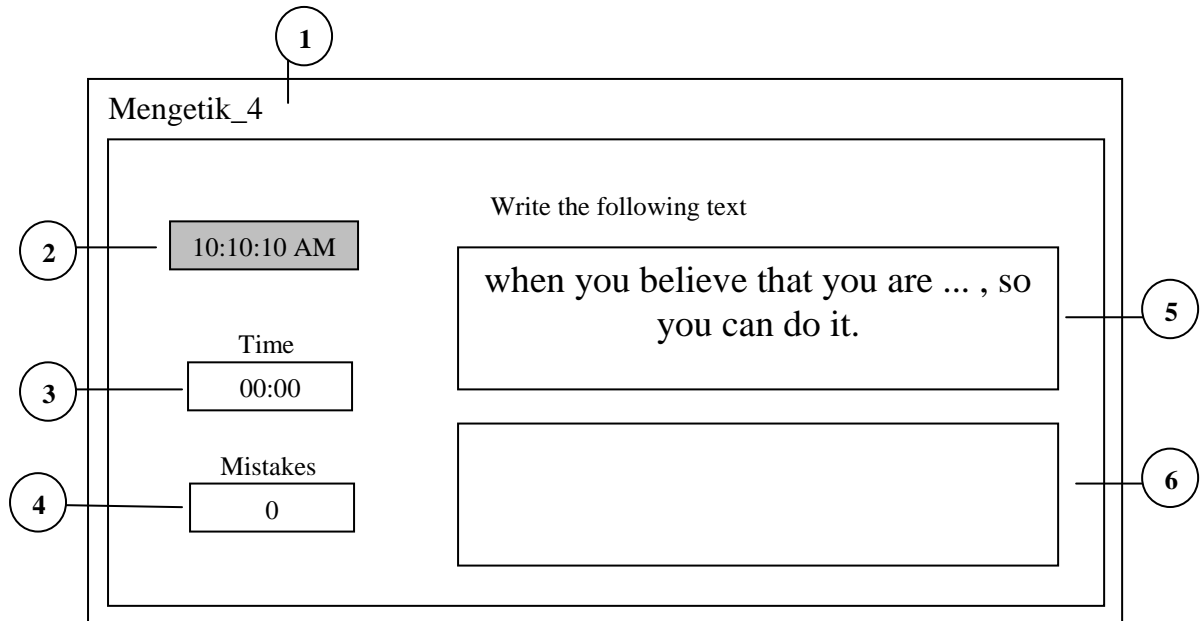


Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title bar</i>
2.	<i>Label</i>	Untuk menampilkan waktu (jam) sesuai dengan format pada PC atau laptop yang digunakan pada saat uji coba program
3.	<i>Label</i>	Untuk menampilkan durasi lamanya waktu yang digunakan <i>user</i> pada saat menggunakan aplikasi
4.	<i>Label</i>	Untuk menampilkan jumlah kesalahan yang dilakukan <i>user</i> pada saat menggunakan aplikasi
5.	<i>Label</i>	Tempat teks yang akan di ikuti
6.	<i>Label</i>	Tempat teks yang akan di ketik

5. Form Mengetik-4

Adapun form mengetik-4 yang dirancang oleh penulis adalah sebagai berikut :



Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title bar</i>
2.	<i>Label</i>	Untuk menampilkan waktu (jam) sesuai dengan format pada PC atau laptop yang digunakan pada saat uji coba program
3.	<i>Label</i>	Untuk menampilkan durasi lamanya waktu yang digunakan <i>user</i> pada saat menggunakan aplikasi
4.	<i>Label</i>	Untuk menampilkan jumlah kesalahan yang dilakukan <i>user</i> pada saat menggunakan aplikasi
5.	<i>Label</i>	Tempat teks yang akan di ikuti
6.	<i>Label</i>	Tempat teks yang akan di ketik

6. Form Mengetik-5

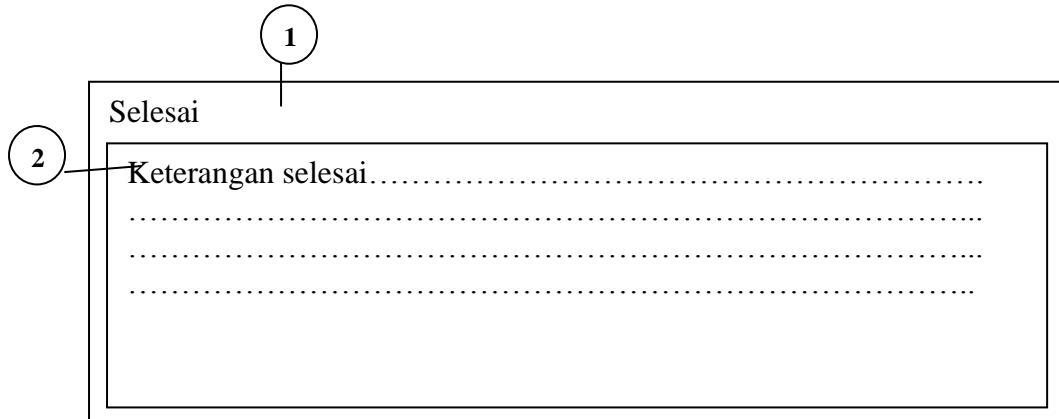
Adapun *form* mengetik-5 yang dirancang oleh penulis adalah sebagai berikut :

Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title bar</i>
2.	<i>Label</i>	Untuk menampilkan waktu (jam) sesuai dengan format pada PC atau laptop yang digunakan pada saat uji coba program
3.	<i>Label</i>	Untuk menampilkan durasi lamanya waktu yang digunakan <i>user</i> pada saat menggunakan aplikasi
4.	<i>Label</i>	Untuk menampilkan jumlah kesalahan yang dilakukan <i>user</i> pada saat menggunakan aplikasi
5.	<i>Label</i>	Tempat teks yang akan di ikuti
6.	<i>Label</i>	Tempat teks yang akan di ketik

7. *Form Selesai*

Form selesai adalah *form* yang berisi tentang keterangan bahwa *user* telah berhasil menyelesaikan permainan latihan mengetik. Adapun rancangan *form* selesai yang dirancang oleh penulis adalah sebagai berikut :

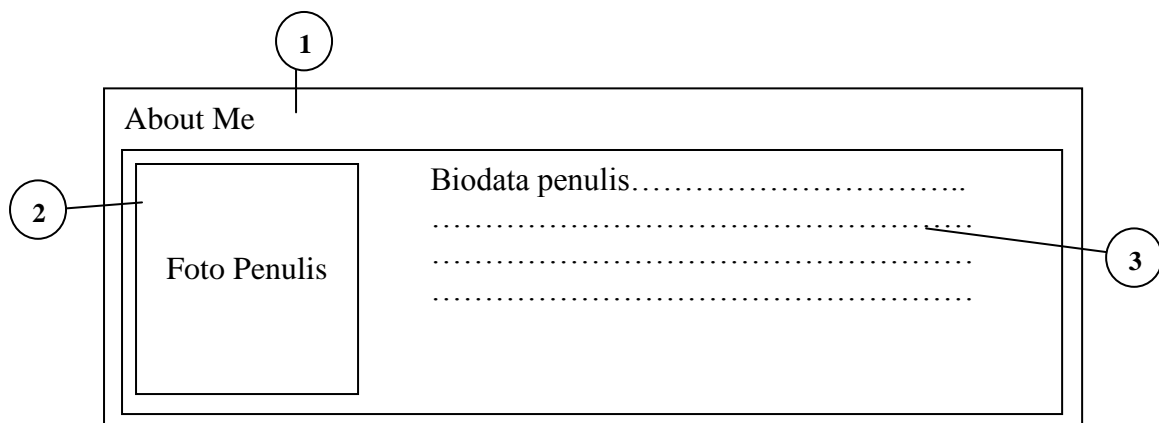


Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title Bar</i>
3.	<i>Label</i>	Keterangan selesai

8. *Form About Me*

Form about me adalah *form* yang berisi tentang biodata *programmer* yang merancang program. Tampilan *form about me* adalah sebagai berikut :



Keterangan:

No	Toolbox	Keterangan
1.	-	<i>Title Bar</i>
2.	<i>Picturebox</i>	Foto penulis
3.	<i>Label</i>	Biodata penulis. Pada saat <i>form about me</i> ditampilkan, maka <i>label</i> (keterangan biodata penulis) menjadi animasi <i>label</i> berjalan dari bawah keatas.

III.5.2. Algoritma

Berikut ini adalah algoritma dari *form* yang dirancang :

1. Algoritma *Form* Mengetik-1 :

```

RESET_TEKS ( siapkan teks pada posisi awal )
  read(T; C)      ( baca karakter awal )
  if C = \.' then
  else
  inisialisasi
  repeat (proses dan baca teks sampai terdeteksi \.')
    pemrosesan terhadap C
    read (T; C)      ( baca karakter selanjutnya )
  until c = \.'
  Terminasi
  Endif

```

2. Algoritma *Form* Mengetik-2 :

```

RESET_TEKS ( siapkan teks pada posisi awal )
  read(T; C)      ( baca karakter awal )
  if C = \.' then
  else
  inisialisasi
  repeat (proses dan baca teks sampai terdeteksi \.')
    pemrosesan terhadap C
    read (T; C)      ( baca karakter selanjutnya )
  until c = \.'
  Terminasi
  Endif

```

3. Algoritma *Form* Mengetik-3 :

```

RESET_TEKS ( siapkan teks pada posisi awal )

```



```

read(T; C)      ( baca karakter awal )
if C = '.' then
else
  inisialisasi
  repeat (proses dan baca teks sampai terdeteksi '.')
    pemrosesan terhadap C
    read (T; C)      ( baca karakter selanjutnya )
  until c = '.'
  Terminasi
Endif

```

4. Algoritma *Form Mengetik-4* :

```

RESET_TEKS ( siapkan teks pada posisi awal )
read(T; C)      ( baca karakter awal )
if C = '.' then
else
  inisialisasi
  repeat (proses dan baca teks sampai terdeteksi '.')
    pemrosesan terhadap C
    read (T; C)      ( baca karakter selanjutnya )
  until c = '.'
  Terminasi
Endif

```

5. Algoritma *Form Mengetik-5* :

```

RESET_TEKS ( siapkan teks pada posisi awal )
read(T; C)      ( baca karakter awal )
if C = '.' then
else
  inisialisasi
  repeat (proses dan baca teks sampai terdeteksi '.')
    pemrosesan terhadap C
    read (T; C)      ( baca karakter selanjutnya )
  until c = '.'
  Terminasi
Endif

```