

LAPORAN AKHIR PROJEK

MATA KULIAH DASAR SISTEM KOMPUTER

WORD COUNTER APLICATION

Dosen Pengampu Mata Kuliah: Ali Tarmuji, S.T., M.Cs.



Disusun Oleh :

Muhammad Dzaky Prayata

2400018119

C (Informatika)

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

2025

KATA PENGANTAR

Puji syukur kami haturkan kehadiran Allah Swt. yang telah melimpahkan rahmat dan hidayah-Nya sehingga saya bisa menyelesaikan project tugas akhir saya, berupa Aplikasi/ Program WORD COUNTER.

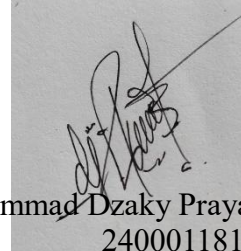
Tidak lupa juga Saya mengucapkan terima kasih kepada semua pihak yang telah turut memberikan kontribusi dalam penyelesaian Project kali ini. Tentunya, tidak akan bisa maksimal jika tidak mendapat dukungan dari berbagai pihak. Terutama penulis mengucapkan rasa terima kasih kepada:

- Bapak Ali Tarmuji, S.T., M.Cs. selaku Dosen pengampu mata kuliah Dasar Sistem Komputer yang memberikan dukungan dan saran kepada penulis.

Sebagai penyusun, Saya menyadari bahwa masih terdapat kekurangan, baik dari penyusunan maupun tata bahasa penyampaian dalam Laporan project ini. Oleh karena itu, saya dengan rendah hati menerima saran dan kritik dari pembaca agar saya dapat memperbaiki laporan project kali ini.

Saya harap semoga Laporan yang saya susun ini memberikan manfaat dan juga inspirasi untuk pembaca.

Yogyakarta, 10 Januari 2025



Muhammad Dzaky Prayata
24000118119

DAFTAR ISI

Table of Contents

LAPORAN AKHIR PROJEK.....	1
KATA PENGANTAR.....	2
DAFTAR ISI	3
A. Ruang Lingkup Aplikasi	4
1. Application Name : Word Counter	4
a. Tujuan Aplikasi	4
b. Fitur Utama.....	4
c. Platform yang Didukung.	4
d. Target Pengguna.	4
e. Teknologi yang digunakan	5
2. Desain/Rancangan Aplikasi	5
3. Hasil Akhir Aplikasi	6
A, KODE PROGRAM.....	6
B. Capture UI dan Penjelasan.....	11
C.Kesimpulan.....	12

A. Ruang Lingkup Aplikasi

1. Application Name : Word Counter

a. Tujuan Aplikasi

Aplikasi Word Counter dibuat dengan tujuan untuk memberikan alat yang mudah digunakan oleh pengguna untuk menghitung jumlah kata dalam sebuah teks. Aplikasi ini dirancang untuk membantu berbagai kebutuhan, seperti penulisan akademik, pembuatan konten, atau kebutuhan profesional lainnya yang memerlukan penghitungan kata secara cepat dan akurat.

b. Fitur Utama

Adapun beberapa fitur yang saya sediakan dalam aplikasi ini, diantaranya :

- a. Penghitungan Kata Secara Real-Time: Pengguna dapat langsung melihat jumlah kata saat teks dimasukkan.
- b. Penghitungan Karakter: Selain jumlah kata, aplikasi ini juga memberikan informasi tentang jumlah karakter dengan atau tanpa spasi.
- c. Dukungan untuk Multibahasa: Aplikasi ini mampu menghitung kata dalam berbagai bahasa, termasuk bahasa yang menggunakan alfabet non-Latin.
- d. Tampilan Ramah Pengguna: Antarmuka yang sederhana dan intuitif untuk memudahkan pengguna dari semua tingkat keahlian.
- e. Mode Input Beragam: Mendukung input teks dari papan ketik dan salinan teks (clipboard).

c. Platform yang Didukung.

Adapun beberapa platform yang dapat mendukung aplikasi ini berjalan diantaranya, windows dan lain-lain.

d. Target Pengguna.

Saya juga menargetkan aplikasi ini dapat digunakan untuk:

- a. Pelajar dan Mahasiswa: Untuk membantu dalam menulis esai, laporan, dan tugas yang memiliki batasan jumlah kata.
- b. Penulis Konten: Membantu penulis artikel, blogger, dan jurnalis dalam panjang tulisan mereka.
- c. Profesional: Bermanfaat untuk pembuatan laporan profesional, mengelola presentasi, dan dokumen bisnis.
- d. Penerjemah: Mendukung penghitungan kata dalam teks multibahasa untuk keperluan pekerjaan mereka.

e. Teknologi yang digunakan

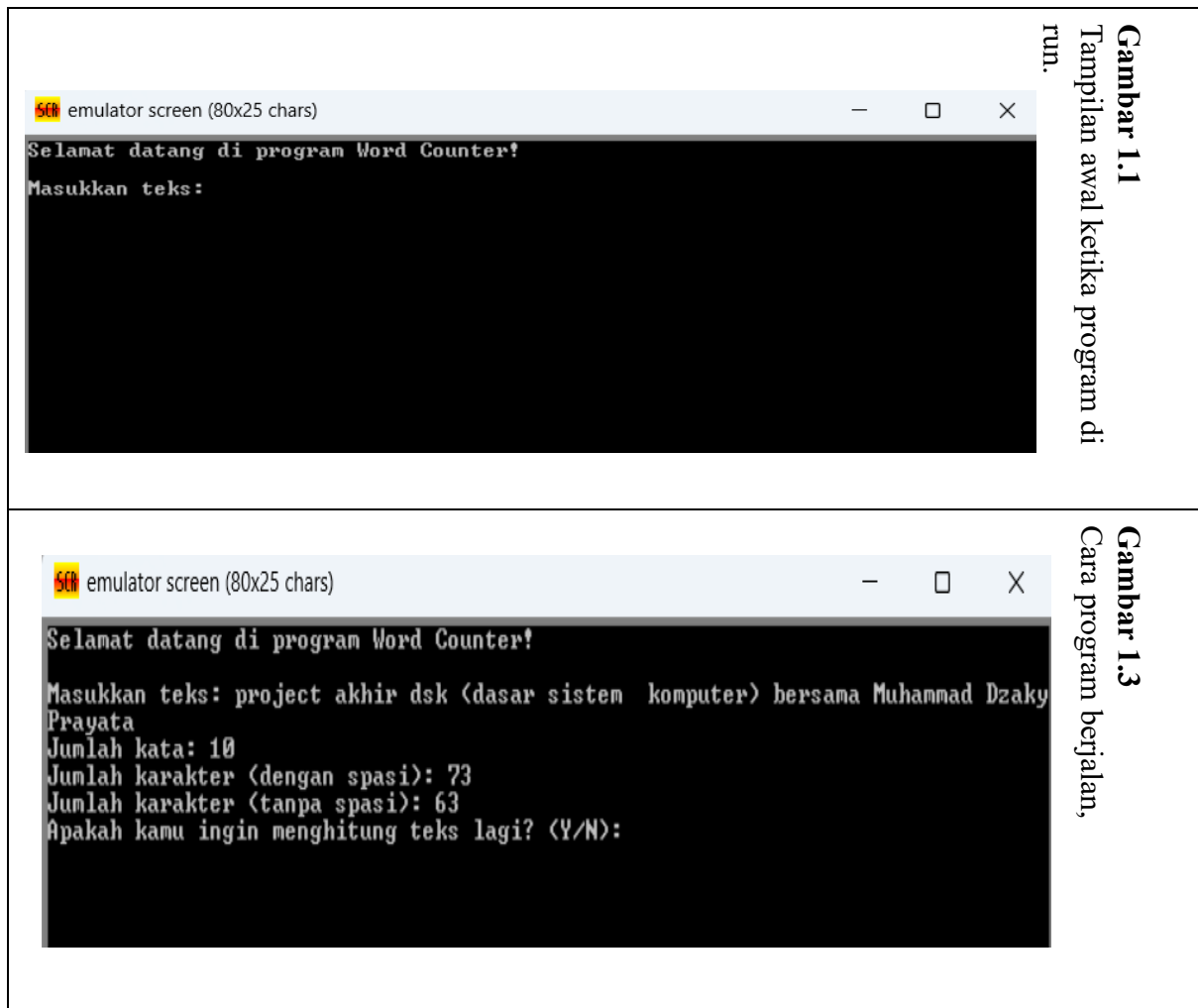
1. Bahasa Pemrograman:

Assembly: Seluruh logika utama aplikasi ditulis dalam bahasa Assembly untuk efisiensi dan kinerja optimal.

2. Platform Pendukung:

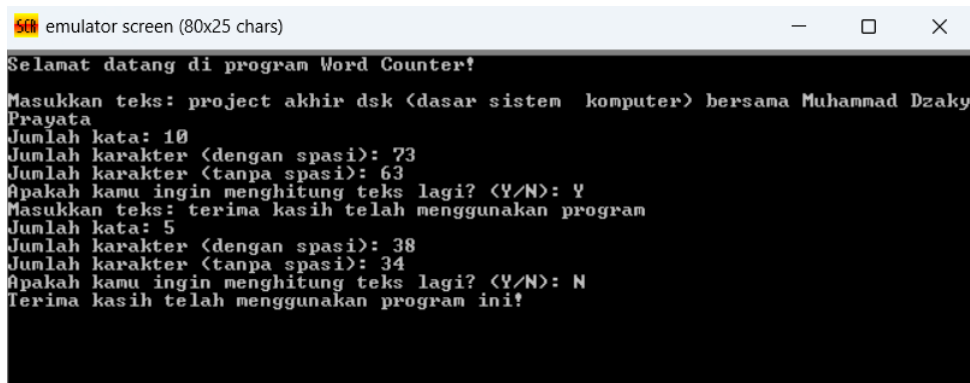
Aplikasi dikembangkan untuk berjalan di lingkungan desktop yang mendukung eksekusi program berbasis Assembl

2. Desain/Rancangan Aplikasi



Gambar 1.3

Apabila program ingin dilanjutkan atau berhenti.

The image shows a screenshot of an emulator window titled "emulator screen (80x25 chars)". The window has standard Windows-style window controls (minimize, maximize, close). The text inside the window is as follows:

```
Selamat datang di program Word Counter!  
Masukkan teks: project akhir dsk <dasar sistem komputer> bersana Muhammad Dzaky  
Prayata  
Jumlah kata: 10  
Jumlah karakter <dengan spasi>: 73  
Jumlah karakter <tanpa spasi>: 63  
Apakah kamu ingin menghitung teks lagi? <Y/N>: Y  
Masukkan teks: terima kasih telah menggunakan program  
Jumlah kata: 5  
Jumlah karakter <dengan spasi>: 38  
Jumlah karakter <tanpa spasi>: 34  
Apakah kamu ingin menghitung teks lagi? <Y/N>: N  
Terima kasih telah menggunakan program ini!
```

Sketsa Desain (Secara visual, ini akan terlihat seperti layar hitam dengan titik yang bergerak sesuai posisi mouse, dan perubahan warna kursor yang dinamis)

3. Hasil Akhir Aplikasi

A, KODE PROGRAM

Kode Program dan penjelasan Program bisa dilihat dibawah :

.model small

.stack 100h

.data

welcome_msg db 'Selamat datang di program Word Counter!', 0Dh, 0Ah, '\$'

prompt db 0Dh, 0Ah, 'Masukkan teks: \$'

output_word_count db 0Dh, 0Ah, 'Jumlah kata: \$'

output_char_count db 0Dh, 0Ah, 'Jumlah karakter (dengan spasi): \$'

output_char_no_space db 0Dh, 0Ah, 'Jumlah karakter (tanpa spasi): \$'

retry_msg db 0Dh, 0Ah, 'Apakah kamu ingin menghitung teks lagi? (Y/N): \$'

invalid_input_msg db 0Dh, 0Ah, 'Input tidak valid, coba lagi.\$'

bye_msg db 0Dh, 0Ah, 'Terima kasih telah menggunakan program ini!', 0Dh, 0Ah, '\$'

input_buffer db 255, 0, 255 dup('\$') ; Buffer input (255 karakter maksimum)

word_count dw 0 ; Penyimpanan sementara jumlah kata

```
char_count dw 0      ; Penyimpanan sementara jumlah karakter dengan spasi
char_no_space dw 0    ; Penyimpanan sementara jumlah karakter tanpa spasi
```

```
.code
```

```
main proc
```

```
    ; Inisialisasi segmen data
```

```
    mov ax, @data
```

```
    mov ds, ax
```

```
    ; Tampilkan pesan selamat datang
```

```
    lea dx, welcome_msg
```

```
    mov ah, 09h
```

```
    int 21h
```

```
    ; Label untuk input ulang
```

```
input_loop:
```

```
    ; Reset penghitung kata dan karakter
```

```
    mov word_count, 0
```

```
    mov char_count, 0
```

```
    mov char_no_space, 0
```

```
    ; Tampilkan prompt untuk input teks
```

```
    lea dx, prompt
```

```
    mov ah, 09h
```

```
    int 21h
```

```
    ; Ambil input teks dari pengguna
```

```
    lea dx, input_buffer
```

```
    mov ah, 0Ah
```

```
    int 21h
```

```
    ; Hitung jumlah kata dan karakter
```

mov si, offset input_buffer + 2 ; Pointer ke awal teks (lewati panjang input byte pertama)

mov bl, 0 ; 0 = di luar kata, 1 = di dalam kata

process_input:

lodsb ; Ambil karakter berikutnya

cmp al, 0Dh ; Akhir input (Enter)

je finalize_counts

inc char_count ; Tambahkan jumlah karakter total

cmp al, ' ' ; Cek apakah spasi

je outside_word

; Jika bukan spasi, tambahkan jumlah karakter tanpa spasi

inc char_no_space

; Jika sebelumnya di luar kata, maka ini awal kata baru

cmp bl, 1

je process_input

mov bl, 1

inc word_count

jmp process_input

outside_word:

mov bl, 0 ; Set flag ke luar kata

jmp process_input

finalize_counts:

; Tampilkan jumlah kata

lea dx, output_word_count

mov ah, 09h

int 21h

mov ax, word_count


```
call print_number
```

```
; Tampilkan jumlah karakter dengan spasi
```

```
lea dx, output_char_count
```

```
mov ah, 09h
```

```
int 21h
```

```
mov ax, char_count
```

```
call print_number
```

```
; Tampilkan jumlah karakter tanpa spasi
```

```
lea dx, output_char_no_space
```

```
mov ah, 09h
```

```
int 21h
```

```
mov ax, char_no_space
```

```
call print_number
```

```
; Tanya apakah ingin mengulang
```

```
retry_input:
```

```
lea dx, retry_msg
```

```
mov ah, 09h
```

```
int 21h
```

```
mov ah, 01h
```

```
int 21h
```

```
cmp al, 'Y'
```

```
je input_loop
```

```
cmp al, 'y'
```

```
je input_loop
```

```
cmp al, 'N'
```

```
je exit_program
```

```
cmp al, 'n'
```

```
je exit_program
```

```
; Tampilkan pesan input tidak valid
lea dx, invalid_input_msg
mov ah, 09h
int 21h
jmp retry_input
```

exit_program:

```
; Tampilkan pesan selamat tinggal
lea dx, bye_msg
mov ah, 09h
int 21h
ret
```

print_number proc

```
; Subroutine untuk menampilkan angka dari register AX
push dx
push cx
push bx

xor cx, cx      ; Reset digit counter
mov bx, 10
```

convert_digits:

```
xor dx, dx
div bx          ; Bagi angka, simpan sisa di DX
push dx         ; Simpan sisa (digit saat ini)
inc cx         ; Tambahkan penghitung digit
test ax, ax
jnz convert_digits
```

print_digits:

```
pop dx          ; Ambil digit terakhir
```

```

add dl, '0'      ; Konversi ke ASCII
mov ah, 02h
int 21h          ; Cetak digit
loop print_digits

pop bx
pop cx
pop dx
ret
print_number endp

end main

```

Diatas adalah kode lengkap dari aplikasi WORD COUNET. Kode ini berfungsi untuk menggambar titik sesuai pergerakan mouse dan mengubah warna kursor berdasarkan posisi X mouse.

B. Capture UI dan Penjelasan

HASIL OUTPUTNYA

- Pengguna akan diminta memasukkan sebuah kalimat yang ingin diidentifikasi
- Kemudian program akan menghitung jumlah kata yang dimasukkan.
- Menampilkan jumlah karakter (dengan spasi), dan karakter (yanpa spasi),
- Perulangan apabila ingin dilanjutkan (Y/N)

```

emulator screen (80x25 chars)
Selamat datang di program Word Counter!
Masukkan teks: project akhir dsk <dasar sistem komputer> bersana Muhammad Dzaky Prayata
Jumlah kata: 10
Jumlah karakter <dengan spasi>: 73
Jumlah karakter <tanpa spasi>: 63
Apakah kamu ingin menghitung teks lagi? <Y/N>: Y
Masukkan teks: terima kasih telah menggunakan program
Jumlah kata: 5
Jumlah karakter <dengan spasi>: 38
Jumlah karakter <tanpa spasi>: 34
Apakah kamu ingin menghitung teks lagi? <Y/N>: N
Terima kasih telah menggunakan program ini!

```

C.Kesimpulan

Program ini adalah penghitung kata berbasis Assembly untuk emulator EMU8086. Fitur utamanya mencakup:

1. Penghitungan kata, karakter (dengan dan tanpa spasi).
2. Interaktif: Pengguna dapat memasukkan teks berulang kali.
3. Dukungan input panjang tanpa batas buffer kecil.

Manfaat dari program ini diantaranya

1. Membantu pemrosesan teks dasar untuk menghitung kata dan karakter.
2. Melatih logika Assembly melalui implementasi pengolahan teks.
3. Efisien dan cocok untuk sistem berspesifikasi rendah.
4. Mudah dikembangkan untuk mendukung fitur tambahan.

Semoga program ini dapat membantu pengguna dan mempermudahnya dalam memahami konsep dasar dalam Bahasa assembly.