

**RESPONSI 2 PRAKTIKUM KONSEP PEMROGRAMAN
MEMBUAT GAME BEJEWEL**



**DISUSUN OLEH:
KELOMPOK 6**

LU'LU'A LIM'A LAILA	L0123076
MOHAMMAD NAZHIIF AL-GHONIY	L0123084
MUHAMMAD ARIF AMIJOYO	L0123089
MUHAMMAD RIFAI AGENG PAMBUDI	L0123098

ASISTEN:

MUHAMMAD ANNYS	L0122102
MUHAMMAD SYAFIQ IBRAHIM	L0122116

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET
2023**

KOMPILASI

```
PS D:\sfml_project\bejeweled> g++ -I C:\SFML-2.6.1\include -c bejeweled.cpp -o bejeweled.o
PS D:\sfml_project\bejeweled>
```

Compile kode bejeweled.cpp menjadi file objek bejeweled.o

Berikut adalah penjelasan singkat tentang setiap argumen dalam perintah tersebut:

- `g++`: Komando untuk mengompilasi program C++.
- `-I C:\SFML-2.6.1\include`: Menyertakan direktori `C:\SFML-2.6.1\include` sebagai tempat mencari file header yang dibutuhkan selama proses kompilasi. Ini berguna jika ada file header yang tidak berada di direktori standar.
- `-c`: Argumen ini memberitahu kompiler untuk menghasilkan objek file (`.o`) daripada mencoba membuat program yang dapat dijalankan langsung.
- `bejeweled.cpp`: Nama file sumber yang akan dikompilasi.
- `-o bejeweled.o`: Menyatakan nama file objek keluaran dari proses kompilasi. Dalam hal ini, nama objeknya adalah bejeweled.o.

```
PS D:\sfml_project\bejeweled> g++ -I C:\SFML-2.6.1\lib -o app.exe -lmingw32 -lsfml-graphics -lsfml-window -lsfml-system -lsfml-main -mwindows
PS D:\sfml_project\bejeweled>
```

Compile file objek bejeweled.o menjadi *executable file*

Berikut adalah penjelasan singkat tentang setiap argumen dalam perintah tersebut:

- `g++`: Komando untuk mengkompilasi program C++.
- `-L C:\SFML-2.6.1\lib`: Menyertakan direktori `C:\SFML-2.6.1\lib` sebagai tempat mencari file library saat proses linking.
- `.\bejeweled.o`: File objek hasil kompilasi dari program utama (bejeweled.cpp).
- `-o app.exe`: Menyatakan nama file keluaran dari proses linking. Dalam hal ini, nama executable file-nya adalah app.exe.
- `-lmingw32`: Menyertakan library MinGW32.
- `-lsfml-graphics -lsfml-window -lsfml-system -lsfml-main`: Menyertakan library SFML untuk grafika, jendela, sistem, dan entry point utama.
- `-mwindows`: Menyertakan opsi ini untuk proyek GUI di Windows agar program berjalan tanpa konsol.

Jika perintah ini dijalankan tanpa kesalahan, akan menghasilkan file `app.exe` yang dapat dijalankan pada sistem Windows.

IMAGE

Berikut ini adalah aset gambar yang digunakan pada game tersebut:

Background



Gems



ANALISIS KODE

```
#include <SFML/Graphics.hpp>
#include <time.h>
using namespace sf;
```

- Kode ini memasukkan header file dari SFML dan time.h.
- Menggunakan namespace sf untuk menghindari penulisan berulang sf:: pada elemen-elemen SFML.

```
int ts = 54; //tile size
Vector2i offset(48, 24);
```

- ts: Ukuran tile (ukuran setiap elemen dalam grid).
- offset: Posisi offset untuk menyesuaikan tampilan grid.

```
struct piece
{
    int x, y, col, row, kind, match, alpha;
    piece() { match = 0; alpha = 255; }
} grid[10][10];
```

- piece: Struktur yang merepresentasikan setiap elemen dalam grid.
- x, y: Koordinat dalam window.
- col, row: Kolom dan baris dalam grid.
- kind: Tipe elemen (gem) yang dipilih.
- match: Jumlah elemen yang cocok dengan elemen ini.
- alpha: Tingkat transparansi elemen (digunakan untuk animasi menghilangkan).

```
void swap(piece p1, piece p2)
{
    std::swap(p1.col, p2.col);
    std::swap(p1.row, p2.row);

    grid[p1.row][p1.col] = p1;
    grid[p2.row][p2.col] = p2;
}
```

- Fungsi untuk menukar dua elemen dalam grid.

```

int main()
{
    srand(time(0));

    RenderWindow app(VideoMode(740, 480),
"Bejeweled!");
    app.setFramerateLimit(60);

    Texture t1, t2;
    t1.loadFromFile("images/background.png");
    t2.loadFromFile("images/gems.png");

    Sprite background(t1), gems(t2);

```

- Menginisialisasi random seed menggunakan fungsi `srand(time(0))`.
- Membuat window SFML dengan nama "Bejeweled!" dan ukuran 740x480 piksel.
- Membuat dua objek Texture dan memuat gambar latar belakang dan gems.
- Membuat dua objek Sprite untuk latar belakang dan gems menggunakan Texture yang telah dimuat.

```

for (int i = 1; i <= 8; i++)
    for (int j = 1; j <= 8; j++)
    {
        grid[i][j].kind = rand() % 3;
        grid[i][j].col = j;
        grid[i][j].row = i;
        grid[i][j].x = j * ts;
        grid[i][j].y = i * ts;
    }

```

- Mengisi elemen-elemen dalam grid dengan nilai-nilai acak.
- Menetapkan koordinat dan jenis elemen pada setiap posisi grid.

```

while (app.isOpen())
{
    Event e;
    while (app.pollEvent(e))
    {
        if (e.type == Event::Closed)
            app.close();

        if (e.type == Event::MouseButtonPressed)
            if (e.mouseButton.button == Mouse::Left)
            {
                if (!isSwap && !isMoving)
                    click++;
                pos = Mouse::getPosition(app) - offset;
            }
    }
}

```

- Melakukan event handling untuk menutup window jika tombol keluar ditekan.
- Meng-handle klik mouse kiri untuk memulai pergerakan elemen.

```

// mouse click
if (click == 1)
{
    x0 = pos.x / ts + 1;
    y0 = pos.y / ts + 1;
}
if (click == 2)
{
    x = pos.x / ts + 1;
    y = pos.y / ts + 1;
    if (abs(x - x0) + abs(y - y0) == 1)
    {
        swap(grid[y0][x0], grid[y][x]);
        isSwap = 1;
        click = 0;
    }
    else
        click = 1;
}

```

```
}
```

- Menangani klik mouse.
- Jika ini klik pertama, mendapatkan koordinat (x0, y0) dari posisi mouse.
- Jika ini klik kedua, mendapatkan koordinat (x, y) dari posisi mouse.
- Jika jarak antara dua klik adalah 1 (sejajar), maka elemen-elemen di posisi tersebut ditukar.

```
//Match finding
for (int i = 1; i <= 8; i++)
    for (int j = 1; j <= 8; j++)
    {
        if (grid[i][j].kind == grid[i + 1][j].kind)
            if (grid[i][j].kind == grid[i - 1][j].kind)
                for (int n = -1; n <= 1; n++)
                    grid[i + n][j].match++;

        if (grid[i][j].kind == grid[i][j + 1].kind)
            if (grid[i][j].kind == grid[i][j - 1].kind)
                for (int n = -1; n <= 1; n++)
                    grid[i][j + n].match++;
    }
```

- Mencari kemungkinan pencocokan (matching) dalam grid.

```
//Moving animation
isMoving = false;
for (int i = 1; i <= 8; i++)
    for (int j = 1; j <= 8; j++)
    {
        piece &p = grid[i][j];
        int dx, dy;
        for (int n = 0; n < 4; n++) // 4 - speed
        {
            dx = p.x - p.col * ts;
            dy = p.y - p.row * ts;
            if (dx)
                p.x -= dx / abs(dx);
        }
    }
```

```

        if (dy)
            p.y -= dy / abs(dy);
    }
    if (dx || dy)
        isMoving = 1;
}

```

- Melakukan animasi pergerakan elemen-elemen dalam grid.

```

//Deleting animation
if (!isMoving)
    for (int i = 1; i <= 8; i++)
        for (int j = 1; j <= 8; j++)
            if (grid[i][j].match)
                if (grid[i][j].alpha > 10)
                {
                    grid[i][j].alpha -= 10;
                    isMoving = true;
                }
}

```

- Melakukan animasi penghapusan elemen yang cocok.

```

//Get score
int score = 0;
for (int i = 1; i <= 8; i++)
    for (int j = 1; j <= 8; j++)
        score += grid[i][j].match;

```

- Menghitung skor berdasarkan jumlah elemen yang cocok.

```

//Second swap if no match
if (isSwap && !isMoving)
{
    if (!score)
        swap(grid[y0][x0], grid[y][x]);
    isSwap = 0;
}

```

- Jika tidak ada pencocokan setelah pertukaran pertama, maka melakukan pertukaran kembali.


```
//Update grid
if (!isMoving)
{
    ...
}
```

- Melakukan pembaruan pada grid setelah animasi pergerakan dan penghapusan selesai.

```
// draw
app.draw(background);

for (int i = 1; i <= 8; i++)
    for (int j = 1; j <= 8; j++)
    {
        piece p = grid[i][j];
        gems.setTextureRect(IntRect(p.kind * 49, 0, 49,
49));

        gems.setColor(Color(255, 255, 255, p.alpha));
        gems.setPosition(p.x, p.y);
        gems.move(offset.x - ts, offset.y - ts);
        app.draw(gems);
    }

app.display();
```

- Menggambar latar belakang dan elemen-elemen dalam grid pada window SFML.
- Melakukan pembaruan tampilan window.